Towards Recursive Models—A Computational Formalism for the Semantics of Temporal Presuppositions and Counterfactuals in Natural Language

Stefano Mizzaro
Department of Mathematics and Computer Science
University of Udine
Via delle Scienze, 206 — Loc. Rizzi — 33100 Udine — Italy
Tel: +39 (432) 55.8456 — Fax: +39 (432) 55.8499
E-mail: `mizzaro@dimi.uniud.it`
WWW: `http://dimi.uniud.it/~mizzaro`

Keywords: artificial intelligence, computational linguistics, natural language processing, semantics, pragmatics, temporal presuppositions, counterfactuals, linguistic knowledge, extra-linguistic knowledge, ontology, content, computational models, recursive models, TOBI.

Edited by:

Received: **Revised:** **Accepted:**

The linguistic phenomena of temporal presuppositions and counterfactuals, situated on the boundary line between semantics and pragmatics, are common to many languages, and the computational treatment of such phenomena is difficult because of their non-monotonic aspect.
These phenomena are presented through a corpus of examples; they are studied emphasizing the various types of knowledge underlying them; and the fragment of language that encloses such phenomena is defined in a way not dependent from a specific language. Then, Recursive Models, a formalism for modeling the semantics of utterances containing temporal presuppositions and counterfactuals, are proposed, described from both functional (by formal specifications) and structural points of view, and compared with related work. Finally, the adequacy of Recursive Models is empirically verified: TOBI (Temporal presuppositions and counterfactuals: an Ontological Based Interpreter), a system that interacts with the user in natural language using the recursive models, is illustrated. TOBI is not based on a deductive system, but uses the more primitive and flexible notion of model-based evaluation; its architecture, flow of control and internal data structures are presented.

1 Introduction

This paper[1] sketches a formalism, named *recursive models*, that can be used for representing, at a semantic-pragmatic level, utterances containing temporal presuppositions and counterfactuals. The power of this formalism is tested by using it in a natural language processing system named TOBI (Temporal presuppositions and counterfactuals: an Ontological Based Interpreter).

The paper is structured in the following way. In Section 2 the linguistic phenomena of temporal presuppositions and counterfactuals are presented and analyzed, and the fragment of natural language relevant for such phenomena is formally defined. Section 3 presents recursive models, the data structures used for modeling the semantics of natural language utterances; such presentation is given from a functional-formal, a structural, and a behavioral point of view. Furthermore, a survey of related work is proposed. Section 4 describes the TOBI system, illustrating its architecture, flow of control, and internal data structures.

---

[1] This work is a revised and extended version of [33, 34].

Section 5 summarizes the work done so far and proposes some future extensions.

2 The language fragment

The linguistic phenomena considered in this work are situated on the boundary between semantics and pragmatics. In the following three subsections: (*i*) a corpus of examples that informally describe such phenomena is presented; (*ii*) the examples are analyzed with respect to a classification of various kinds of knowledge; and (*iii*) a formal (syntactical) definition of the fragment of the language studied is presented.

2.1 The linguistic phenomena

To completely understand the meaning of each utterance, it is important to analyze its relations with the other utterances in the discourse. Following Gazdar [18], an utterance *implies* another utterance if the latter is a consequence of the former (here I give no formal definition of implication). For example, utterance (1)

"Mary met John before she left" (1)

(utterances are enclosed in double quotes) implies utterances (2) and (3):

"Mary met John" (2)

"Mary left." (3)

A particular case of implication between utterances is *entailment*: utterance (1) entails (2). However, entailment is not the only type of implication, as utterance (3) proves: the relation between (1) and (3) is not an entailment, as shown by the fact that the utterance

"Mary met John before she left and he persuaded her to stay at home" (4)

is consistent. If we admit that (3) is entailed by (1), then (3) is also entailed by (4). But (4) entails

"Mary did not leave"

which contradicts (3). Utterance (3) is a *(temporal) presupposition* of (1) [18, 23, 24, 27, 29, 30]. A presupposition is a form of implication weaker than entailment: the second part of (4), *asserting* that Mary stayed at home, *cancels* the presupposition, so we do not have a contradictory utterance.

It is important to remark that although the event 'Mary left' did not happen, it is used in (4) to date the event 'Mary met John'. Moreover, from a logical point of view it seems more correct to say

"Mary met John before she *did not* leave and he persuaded her to stay at home" (5)

instead of (3), but no human would do so. In other words, the problem is in *nonmonotonicity*: utterance (1) implies (3) only *by default* and the second part of (4) deletes the default. Then, entailment can be seen as a *certain* inference, while presupposition as a default (and so *uncertain*) one.[2] Therefore, a system handling such phenomena must be nonmonotonic. The most widely used formalism for this purpose is represented by nonmonotonic logics [11, 21]; however, the approach followed in this work is different, as will be shown later.

It has to be noted that 'after' is not the symmetric counterpart of 'before', as shown by the fact that in the utterance

"Mary met John after she left"

the leaving event cannot be deleted, as done in utterance (4): the utterance

"Mary met John after she left. She did not leave"

is clearly inconsistent.

Furthermore, relationships between events are necessary for example to explain the utterance

"Mary left before meeting John", (6)

---

[2] Note that there are two different views of temporal presuppositions (and of presuppositions in general). On the one side (what might be called an *a priori* view) they are necessary for giving a truth value to the whole sentence: the name 'presupposition' comes from here. On the other side (*a posteriori* view) they leave a trace as a defeasible inference: as it was pointed out before, temporal presuppositions can be seen as a kind of implication weaker than entailment. Here I am interested in the latter aspect; the former is analyzed in a lot of works [23, 24, 29].

in which the meeting event is presupposed, but it is immediately deleted on the basis of *world knowledge*; so the leaving event prevents the meeting.

Another linguistic phenomenon strictly related with the previous ones is that of (*conditional*) *counterfactuals*. In fact, (4) implies:

"If Mary had not met John, she would have left", (7)

that is used for referring to an hypothetical course of events, or a *non-real world* (the world in which Mary did not meet John). It is important to observe the (perhaps unexpected) fact that the meaning of an utterance as "α before β" is sometimes more similar to an utterance of type "if not α', β'" (where α' stays for the subjunctive form of α and β' for the conditional form of β) than to an utterance of kind "β after α".

Two other related linguistic phenomena have been considered. The first one is exemplified by

"The bullet deviated before hitting Mary. Nevertheless it hit her." (8)

What happens in this utterance can be explained in the following way:

- analogously to utterance (6), it is presupposed that the bullet hit the target, but such presupposition is immediately deleted on the basis of world knowledge: human beings know that if a thing is deviated from its trajectory, usually it does not hit the original target;
- in the second part of the utterance it is asserted that the bullet hit the target anyway. To do this, it is not correct to use the conjunction 'and' as done in (4) to cancel a presupposition. A more powerful way, the use of the conjunction 'nevertheless', is needed. The reason is that what has to be deleted in this case (the non-occurrence of the non-hitting, derived from world knowledge considerations) is something 'stronger' than the temporal presupposition of (4).

The second phenomenon is shown by the following utterance, implied by (8):

"Even if the bullet had not deviated, it would have hit Mary." (9)

Such 'even if' utterances (that I shall call *weak counterfactuals*) play, concerning 'nevertheless' utterances (i.e. utterances like (8) above), the same role that usual counterfactuals have in the case of 'and' sentences. That is, utterance (9) is for (8) what (7) is for (4).

The standard treatment of temporal presuppositions [18, 23, 24, 27, 29, 30] is not entirely satisfactory: there is no deep explanation of why 'before' should introduce a presupposition, while 'after' should introduce an entailment. The point is that an *ontology of time* is not taken into account: time is ordered and the future unknown and partially unpredictable, and these facts must be taken into account when dealing with utterances containing 'before' and 'after'. In this way, no *linguistic* explanation of why an event introduced by 'before' can be deleted and one introduced by 'after' cannot is required. Linguistically, one can—and ought—only say that secondary sentences started by 'after' and 'before' introduce a presupposition that can be deleted later. The explanation of the asymmetry between 'before' and 'after' must be found at a deeper level, in the way we, human beings, perceive and treat the time. This point is investigated in the next section.

2.2 Linguistic and extra-linguistic knowledge

It is common usage [29] to divide the knowledge utilized for making inferences about an utterance into two classes: *linguistic knowledge* (LK) and *world knowledge* (WK). In this section I propose a more subtle distinction, that will be useful for both understanding and treating the linguistic phenomena at hand.

First, it is possible to distinguish between LK and *extra-linguistic knowledge* (ELK). LK is used for deriving facts from an utterance through pure linguistic rules.[3] For instance: a proper noun

---

[3] Let us note that 'knowledge' and 'inference' can be defined from the standpoint of mathematical logic. A formal calculus [16] is made of axioms (that represent known facts about a domain) and inference rules (that model the inference process). Starting from the axioms, and using inference rules, one can derive (infere) other facts. The axioms may be divided into groups corresponding to different kinds of knowledge involved. In the same way, also the inference rules may be grouped. Inferences and derived facts can be classified according to the kind of the axioms

stands for an individual; if a noun phrase is plural, then it denotes more than one individual; if the tense of a verb is ‘simple past’ (‘future’), then the event that it denotes happened in the past (will happen in the future); if an event is described in the main (secondary) proposition, then it is entailed (presupposed); and so on. ELK inferences instead are not directly derived from the utterance through linguistic considerations, but from other knowledge sources (i.e. from the world as we know it): a human proper noun like ‘Mary’ usually denotes a female human being; if someone is dead, he cannot do anything; if an event happened in the past, it cannot be modified; if an event is expected to happen in the future, it may or may not happen; and so on. The distinction between LK and ELK is not so clear-cut, being sometimes difficult (or arbitrary) to classify an axiom or an inference. Anyway it is interesting to study how far it is possible to push this dichotomy.

Second, both the LK and ELK inferences and derived facts can be *uncertain* or *certain*. The uncertain LK inferences were called in the previous section ‘presuppositions’, the certain ones ‘entailments’. Another kind of uncertain LK inferences are *implicatures* [30]. ELK inferences are often uncertain (the ‘real’ world is very difficult to model: the research on WK, or *common sense* [14, 25] is one of the main subfields of artificial intelligence): ‘Mary’ usually denotes a female human being, but it might denote a hurricane, or a boat, or something else; if a bullet is deviated, usually it does not hit the target, but sometimes this could happen anyway; and so on. But ELK inferences can also be certain: if an event happened in the past it cannot be modified; if an event is said to happen in the future, it might happen or not happen; and so on. In the following I will call *ontology* the certain ELK and *content* the uncertain ELK. Informally speaking, ontology is the component of knowledge that has a general logical status; on the contrary, content is the component of knowledge that is highly situation dependent.

Let us consider a concrete example. In Table 1 some of the facts that can be derived from utterance (8) reported here below are shown and classified along the LK/ELK and uncertain/certain dimensions.[4]

LK	Uncertain (presupposition)	An event of ‘hitting’ (from the before-clause) happened in the past
	Certain (entailment)	‘The bullet’ and ‘Mary’ denote individuals
		An event of ‘deviation’ happened in the past
		The deviation-event happened before the hitting-event
		An event of ‘hitting’ (from the nevertheless-clause) happened in the past
ELK	Uncertain (content)	The individual denoted by ‘Mary’ is a female human being
		The hitting-event, because of the deviation-event, did not happen
	Certain (ontology)	The hitting-event is in the future for what concerns the before-clause, so it is uncertain.

Table 1: Inferences from utterance (8).

“The bullet deviated before hitting Mary. Nevertheless it hit her.” (8)

The phenomenon of temporal presuppositions seems to be an expression of the ontology of time, not of the content of time. The ontology of time is its *ordering* and the fact that while the past is in a sense *closed*, the future is *open*. This leads to certain inferences. On the other side, the *metric* of time is a content characteristic, in that the subjective evaluation of the duration of a time interval may vary depending on the situation, and this usually leads to uncertain inferences. Then, the phenomenon of temporal presuppositions can be explained in the following way: an event in the future cannot be certain, because of the ontology

---

and inference rules used. Therefore, it is possible to speak of axioms (inference rules, inferences, and facts) of LK and ELK type. Examples of distinctions can be, besides the WK/LK in [29], the terminological/assertional [9], or the symbolic/subsymbolic [38] dichotomies.

[4] The case of the certain ELK inference might seem a bit awkward. A more convincing example is the fact that in utterance “Mary met John after she left” the leaving event did certainly happen.

of time (partial unpredictability of the future).[5] This is why 'before' introduces a temporal presupposition, while 'after' does not.

In this work, I am interested in those parts of LK and ELK that are related with temporal presupposition and counterfactuals. The content is not the focus of this research, but it plays a role (indeed a marginal one) into the above described linguistic phenomena. As a matter of fact, content inferences can contradict presupposed and/or entailed events, thus sometimes (but only if relevant and necessary) it will be necessary to take content into account. Note that entailments overcome content inferences (as, for instance, in utterance (8)), and that content inferences overcome presuppositions (as, for instance, in (4), (6) and (8)).

2.3 Abstract syntax

In order to analyze the above introduced phenomena, it is sufficient to work on a restricted language fragment, defined in this section. The usual way to formally define a fragment of the language is to provide a *grammar*. Since the considered phenomena occur in many natural languages (almost every western language has the syntactic constructs necessary for expressing the previous utterances), I prefer here a more abstract description, to some extent independent from the particular language adopted. I shall call such formalism *abstract syntax*.

The first step to define the abstract syntax of the relevant natural language fragment (that will be denoted with L) is to specify a family of *syntactic functions*, functions that syntactically manipulate sentences of the natural language to obtain other sentences. The definition of the abstract syntax of L is then obtained by means of a set hierarchy: starting from a set of simple sentences, other sets containing *complex* and *compound* sentences [42] are obtained as the range of syntactic functions. The union of these sets will be L.

The syntactic functions used to cover all the linguistic phenomena presented in the previous section are the following:

- $neg(s)$: returns the negation of sentence s. For example, if s is 'Mary left' (sentences are enclosed in single quotes), $neg(s)$ is 'Mary did not leave';

- $before(s_1, s_2)$: returns the complex sentence formed by the main clause s_1 and the temporal subordinate s_2, introduced by 'before'. Observe that the syntactic functions do not only concatenate the strings given as arguments, but also (syntactically) manipulate them to obtain the correct result. For example, from 'Mary met John' and 'Mary left', using the syntactic function *before*, one should obtain 'Mary met John before she left' and not 'Mary met John before Mary left';

- $after(s_1, s_2)$: returns the complex sentence formed by the main clause s_1 and the temporal subordinate s_2, introduced by 'after';

- $and(s_1, s_2)$: returns the compound sentence constituted by the two sentences s_1 and s_2 joined by the conjunction 'and';

- $nevertheless(s_1, s_2)$: returns the compound sentence constituted by the two sentences s_1 and s_2 joined by the conjunction 'nevertheless'. Usually, $nevertheless(s_1, s_2)$ is a pair of sentences separated by a full stop. Here this detail is not important, in that the two sentences, from a semantic point of view, are co-ordinated;

- $cf(s_1, s_2)$: returns the counterfactual sentence with s_1 as antecedent and s_2 as consequent. For example, if s_1 is 'Mary met John' and s_2 is 'Mary left', $cf(neg(s_1), s_2)$ is 'If Mary had not met John, she would have left';

- $wcf(s_1, s_2)$: returns the weak counterfactual, i.e. a sentence (syntactically) differing from a counterfactual one in that 'even if' substitutes 'if'. For instance, if s_1 and s_2 are the two sentences just met for the *cf* function, then $wcf(neg(s_1), s_2)$ is 'Even if Mary had not met John, she would have left'.

Now, the syntactic functions listed above are used to formally define the fragment L: as it was said before, a set hierarchy is built, the last set

---

[5] It is important to point out that 'future' refers to the *point of reference*, not to the *point of speech* [36]. In utterance (1), both the events happened in the past ('met' and 'left'), but the second is in the future of the point of reference.

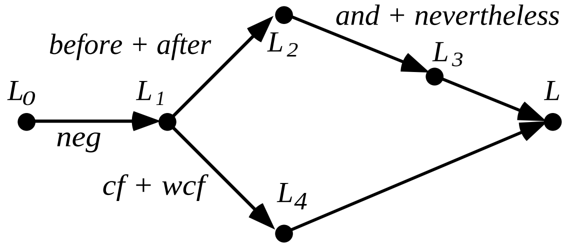

Figure 1: The construction of L.

of the hierarchy being L. The process of L's construction is illustrated in Figure 1. Nodes indicate the subsets of L, arcs mean set inclusion and arc labels show which syntactic functions are used to obtain the following sets.

The first set of the hierarchy, L_0, contains simple sentences, like 'Mary met John' or 'Mary left', and so on.

Using the *neg* function, the set L_1 can be defined as[6]

$$L_1 = L_0 \cup \mathit{neg}(L_0).$$

L_1 contains sentences and their negations, so sentences as 'Mary did not meet John' belong to L_1.

The next set is defined by the *before* and *after* functions:

$$L_2 = L_1 \cup \mathit{before}(L_1, L_0) \cup \mathit{after}(L_1, L_1).$$

Note that the temporal clauses introduced by 'before' are always affirmative, as observed in [24] and as indicated by the utterances and sentences (in particular (5)) presented above.

The following steps are:

$$\begin{aligned} L_3 &= L_2 \cup \mathit{and}(L_2, L_1) \cup \mathit{nevertheless}(L_2, L_1), \\ L_4 &= L_1 \cup \mathit{cf}(L_1, L_1) \cup \mathit{wcf}(L_1, L_1). \end{aligned}$$

The final set, L, is then obtained as

$$L = L_3 \cup L_4.$$

In this section, only sentences have been dealt with, but the extension to the case of utterances is immediate. In fact, if u is an utterance, then u is a pair (s, c), where s is the sentence and c the context. Then,

$$\mathit{neg}(u) = (\mathit{neg}(s), c)$$

and similarly for the other syntactic functions.

3 Recursive models

This section presents *recursive models* (RM), a formalism that can be used to represent the meaning of utterances at a semantic/pragmatic level. In Section 3.1 the RMs are defined as an instance of the class of computable models. In Section 3.2 RMs are seen as an abstract data type, whose formal specifications are given. In Section 3.3 the structure of RMs is described. In Section 3.4 the functions that build and use an RM are analyzed and a possible implementation is sketched. In Section 3.5 related work is discussed.

3.1 Computable models

From a computational perspective, two approaches are possible for representing the semantics of a discourse,[7] and for using such representation in finding implications between the discourse and following utterances. In the first, 'inferential', approach, the discourse is translated into a theory (a set of logical formulas) Γ; the same happens to a following utterance, obtaining, say, the logical formula ϕ; then, to discover whether the discourse implies the utterance, an inference procedure $\vdash$ is used for testing whether $\Gamma \vdash \phi$.[8]

In the second, 'model-theoretic', approach, the discourse is used to build a model M, and an evaluation function (usually denoted by $\models$ in mathematical logic) is used in order to test whether $M \models \phi$.

These are obviously two quite different approaches: in the former the central notions are a set of axioms (to which further ones can be added for taking into account new utterances) and a set of inference rules; the latter is based on the two

---

[6] Here and in the following of this section, the standard notation for using sets as functions arguments is used: $\mathit{neg}(L_0)$ stands for $\{\mathit{neg}(s) \mid s \in L_0\}$, and similarly for the other syntactic functions, paying attention to their arity.

[7] A *discourse*, or a *text*, can be defined as a sequence of utterances. The concepts of implication, entailment and presupposition described in Section 2.1 can be extended in a natural way in order to deal with discourse.

[8] For an explanation of the concepts derived from mathematical logic, see for instance [12, 16].

functions that, respectively, *integrate* (*int* in the following) a previous model with the information of a new utterance, and *evaluate* (*eval* in the following) an utterance in a previously built model.

If the representation of the semantics of a discourse has to be used by an algorithm, both these approaches reveal some decidability problems. In the inferential approach, this happens when neither the utterance (ϕ) nor its negation ($\neg\phi$) are an entailment of the discourse (Γ), and this is a common situation, in that the logical theory Γ is not necessarily *complete*. The standard solution is to abort the inference process when it is too long, the length of the process being the number of inference steps or the computation time. In the model-theoretic approach, similar decidability problems arise when the evaluation function is not computable. This leads to a constraint on the models: their expressivity has to be sacrificed, for obtaining a computable *eval* function. I shall call the models with such property *computable models.*

In the next subsections I will propose an instance of computable models named *recursive model* (RM) that can be used to represent the semantics of utterances belonging to the language fragment defined above. I will not formally prove the computability of the corresponding *eval* function; instead, the approach is empirically tested by utilizing RMs in a system whose implementation will be described in Section 4.

3.2 Formal specifications of recursive models

This section describes the RMs from a functional point of view, formally specifying their behaviour without referring to their structure. In other words, I propose the *formal specifications* of the Abstract Data Type (ADT) RM. The formal specifications of the ADT RM being rather complex, only a brief sketch is presented here. I will define (some of) the *sorts*, (some of) the functions that define the ADT RM, toghether with their *signature*, and (some of) the *axioms* that describe the behaviour of the functions.[9]

The sorts of ADT RM are:[10]

- U, the set of all utterances. On this sort, all the syntactic functions presented in Section 2.3 are assumed to be defined;
- M, the set of all RMs;
- B, the set of boolean values ($\{true, false\}$). I assume that the usual logical connectives are defined as functions on this sort;
- Bu, the set obtained adding the undefined value to the set of boolean values ($\{true, false, undef\}$). Also on this sort I assume that some logical operations are predefined. There exist various 3-valued logics; among them I need Bochvar's logic [8, 40], in which the *undef* value is 'contagious' (i.e., if *undef* is one of the arguments of a logical operation, the result will be *undef* too).

On such sorts, the following functions are defined (together with the signature of the functions, I also present an informal description of their behaviour):

- *create:* $\rightarrow M$, that returns an empty RM;
- *int:* $U \times M \rightarrow M$, that returns a new RM obtained integrating the information of a new utterance in a previously existing model;
- *eval:* $U \times M \rightarrow Bu$, that evaluates the truth value of an utterance in a model;
- *modify:* $U \times M \rightarrow M$, that, given a counterfactual utterance and an RM as arguments, returns the RM obtained modifying the original RM in such a way that the antecedent of the counterfactual utterance is evaluated *false*. The model obtained is named *counterfactual model*;
- *pref:* $2^M \rightarrow M$, that selects the preferred RM among the set of plausible ones. For example, in the case of utterance (1), *pref* should choose the RM in which Mary left, and not

---

[9] Note that I said 'formal', not 'algebraic' specifications: in algebraic specifications [7, 39] the axioms must be equations, in order to have an executable object. Here I am interested only in obtaining a formal definition of the behaviour of the ADT RM, not in the computational aspect (that will be tackled in the following), so I prefer not to have restrictions on the shape of axioms.

[10] These are not all the sorts needed to completely specify the ADT RM. Another sort, the set E of all events, on which the functions that describe the causal links between events must be defined, is necessary.

the one in which Mary did not leave. This function, together with the following three, is needed because of the nonmonotonic aspect of the phenomenon of temporal presuppositions;

- *contr:* $U \times M \to B$, that is *true* iff an utterance, once integrated in an RM, leads to a contradiction. This happens, for example when integrating the second part of (4) in the RM obtained from (1), where it is not longer true that Mary left;
- *rev:* $U \times M \to M$, that operates a revision of an RM when it, together with an utterance, leads to a contradiction;
- *intmon:* $U \times M \to M$, that can integrate utterances that do not present contradiction with the existing RM. Therefore, *intmon* cannot treat nonmonotonicity, but it is the core of *int* function;
- *intset:* $U \times 2^M \to 2^M$, that from the set of previous plausible RMs and an utterance returns another set of RMs. This function is needed because it is possible to build more than one RM from an utterance, as is shown, for example by utterance (1) and (4);
- *evalset:* $U \times 2^M \to Bu$, that is *true* iff the utterance given as the first argument is evaluated *true* in all the RMs belonging to the set given as the second argument;
- *entail:* $U^* \times U \to Bu$, that is *true* iff an utterance is an entailment of a discourse, i.e. a sequence of utterances (with the * operator I indicate the concatenation of utterances);
- *imply:* $U^* \times U \to Bu$, that is *true* iff an utterance is evaluated *true* in the preferred model of a discourse. These two last functions can be defined in terms of the previous ones, see below.

As it was said, I present here only some examples of the axioms needed for the ADT RM. One of such axioms defines the *int* function using functions *intmon*, *rev* and *contr*:[11]

$$\begin{aligned} \mathit{int}(u,m) \;\;=\;\; & \textbf{if } \mathit{contr}(u,m) \\ & \textbf{then } \mathit{intmon}(u, \mathit{rev}(u,m)) \\ & \textbf{else } \mathit{intmon}(u,m). \end{aligned}$$

The evaluation of counterfactual and weak counterfactual utterances takes place in a peculiar way. A counterfactual utterance $\mathit{cf}(u_1,u_2)$ is evaluated *true* if and only if its antecedent u_1 and consequent u_2 are evaluated *false* and the event represented in the consequent should have happened if the event in the antecedent had happened. In other words, the evaluation of $\mathit{cf}(u_1,u_2)$ in an RM m takes place evaluating u_1 and u_2 in m and then evaluating u_2 in a model obtained *modifying* the RM m on the basis of the antecedent u_1. A weak counterfactual $\mathit{wcf}(u_1,u_2)$ behaves in the same way, with the exception that the consequent u_2 must be evaluated *true*. This is formalized by two axioms:

$$\begin{aligned} & \mathit{eval}(\mathit{cf}(u_1,u_2),m) = \\ & \quad \textbf{if } \mathit{eval}(u_1,m) = \mathit{false} \textbf{ and} \\ & \quad\quad \mathit{eval}(u_2,m) = \mathit{false} \\ & \quad \textbf{then } \mathit{eval}(u_2, \mathit{modify}(u_1,m)) \\ & \quad \textbf{else } \mathit{false} \end{aligned}$$

$$\begin{aligned} & \mathit{eval}(\mathit{wcf}(u_1,u_2),m) = \\ & \quad \textbf{if } \mathit{eval}(u_1,m) = \mathit{false} \textbf{ and} \\ & \quad\quad \mathit{eval}(u_2,m) = \mathit{true} \\ & \quad \textbf{then } \mathit{eval}(u_2, \mathit{modify}(u_1,m)) \\ & \quad \textbf{else } \mathit{false} \end{aligned}$$

(note the use of the syntactic functions *cf* and *wcf*).

Another axiom defines the *imply* function in terms of *eval* and *int*:

$$\mathit{imply}(\mathbf{u_1},u_2) \;\;=\;\; \| \, \mathit{eval}(u_2, \mathit{int}(\mathbf{u_1}, \mathit{create}())) \, \|,$$

where the symbol $\mathbf{u_1}$ denotes a sequence of utterances, i.e. a discourse, and the symbol $\| \; . \; \|$ indicates Bochvar's 'assertion operator', that maps the *undef* value in *false* and does not affect the other two logic values.[12]

A similar axiom can be given for the definition of the *entail* function. Here the notion of *set of*

---

[11] The if-then-else operator used here has to be intended as a declarative one, without any procedural meaning.

[12] Note that the first argument of *int* is a sequence of utterances, while *int* should have as argument a single utterance (*int:* $U \times M \to M$). But it is easy to define by

models must be used: an utterance is entailed by another utterance only if the former is *true* in all the models of the latter. Such an axiom is:

$$entail(\mathbf{u_1}, u_2) = \\ \| evalset(u_2, intset(\mathbf{u_1}, \{create()\})) \| .$$

As a last example, the following axiom defines the connection among the *int*, *intset* and *pref* functions:

$$int(u, m) \quad = \quad pref(intset(u, \{m\})).$$

The meaning of this axiom should be clear: the RM obtained by *int* is the preferred one in the set of *all* plausible models, as generated by the *intset* function.

3.3 Structure of recursive models

The previous section has shown how to formally define the properties that RMs must have. Here, the *structure* of the RMs is presented.

Roughly speaking, an RM is constituted by *instances* of classes of an encyclopedia and *relations* among those instances. Therefore, an *encyclopedia* is needed, that is a taxonomy of *categories* and *concepts*. The encyclopedia is a knowledge base, and is needed in order to know that Mary and John are persons, hence living beings, and so on; that the meeting of Mary and John is an event, etc.

Using the operation of *instantiation* it is possible to create a *token* for each individual mentioned in the utterance. Referring to utterance (1), there will be tokens for 'Mary', 'John' (instances of the class `person`), 'met' and 'left' (instances of the class `event`). Every token has an associated identifier; I shall use uppercase letters for instances of objects (`M` for 'Mary', `J` for 'John'), and lower case letters for events (`m` for 'met', `l` for 'left', etc.). As usual, tokens inherit *slots* from their parent concepts, so `M` is the value of the slot `agent` of `m` and `J` is the value of the slot `theme` of `m`. Moreover, between tokens `m` and `l` there is a temporal *relation* to indicate that the meeting took place before the leaving.

Tokens, slots and relations are not sufficient to obtain a complete RM, since by using only these components, one would obtain the same RM for the utterance

> "Mary did not meet John before she left"

and this is clearly a problem. To deal with event occurrence and object existence, other elements are introduced in the RM: *spaces*, *attachments* and *signs*.

A *space* is needed because not only an object exists, or an event takes place; it is more correct to say that an object exists (or an event takes place) *in a world*. Consider utterance (4): Mary did not leave in the *real world*, but it is correct to say that Mary left in the counterfactual world (see utterance (7)) in which she did not meet John. Analogously, it is possible to say that Donald Duck does not exist in the real world, but he exists in Walt Disney's world.

So, a space is a formal tool for representing alternative worlds. I indicate the real world with `[]`. It is possible to represent the object existence and the event occurrence *attaching* every token to the right world: the relation between token and world is named *attachment*. Finally, attachments are labelled with a *sign* in order to deal with non-existence and non-occurrence, both of which are represented by a negative sign, whereas a positive sign obviously means existence and occurrence.

As illustrated in Section 2.1, the occurrence of an event may be certain (the meeting of (1)) or uncertain (the leaving of (1)); this can be dealt with using *certain* and *uncertain* signs. In the RM of (1), the signs labelling the attachments of the tokens for 'met' and 'left' are both positive, but only the first is certain, while the second is uncertain.

The RM obtained for (1) is illustrated in Figure 2. Only the portion of the encyclopedia needed to build the RM of the utterance is represented (in the upper gray area, while in the lower white area, the proper RM is sketched): each rectangle stands for a concept. The relations *is-a* (between two concepts) and *instance-of* (between a concept and a token) are represented by labelled grey arcs, tokens are shown as circled

---

recursion int': $U^* \times M \to M$ in the following way:

$$\begin{aligned} int'([], m) &= m; \\ int'([u_1|\mathbf{u_2}]) &= int'(\mathbf{u_2}, int(u_1, m)) \end{aligned}$$

(where the standard symbology of Prolog lists is used in order to indicate a sequence of utterances) and redefine *int* as int'. The same remark has to be made for the *intset* function in the following equation.

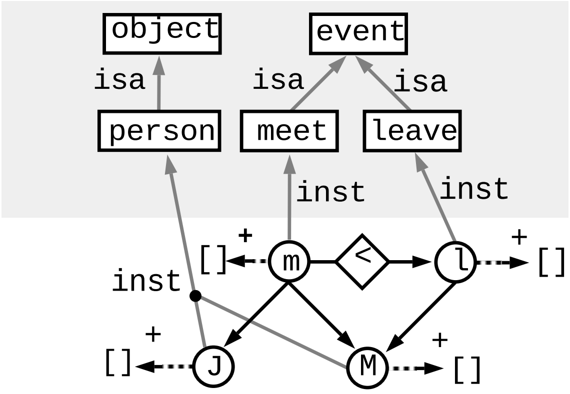

Figure 2: Graphic representation of the RM of the utterance (1).

letters, slots are illustrated by means of oriented arcs, relations, as usual in entity-relationship diagrams used in data base theory [13], are represented by arcs labelled with a rhombus (the symbol `<` stands for 'precedes temporally'), a dashed arc represents an attachment, a bold sign is certain and a plain text sign is uncertain. For the sake of simplicity, in the graphic representation the names of the slots are not illustrated.

The RM in Figure 2 models the meaning of (1). Nevertheless, there is another element to add for dealing with the *causal links* relating the occurrence (or non-occurrence) of events. Examples can be found in utterances (4) and (7) (the occurrence of the meeting with John causes the occurrence of the event 'Mary stayed at home') and (6) (the occurrence of Mary's leaving causes the non-occurrence of the meeting with John).

The elements used in RMs to represent such causal relations are named *justifications*, and are represented by curved arcs. As signs, justifications may also be certain or uncertain. In order to understand the role of these new elements, consider Figure 3, in which the RM of (4) is represented. Here and in the following, for the sake of simplicity, I have omitted the representation of the encyclopedia (i.e. the classes and the `isa` and `inst` relations): the letters labelling the tokens should be sufficient for understanding which class each token is an instance of. Furthermore, the token `p` is assumed to be an instance of the ad-hoc class `persuade to stay at home`.

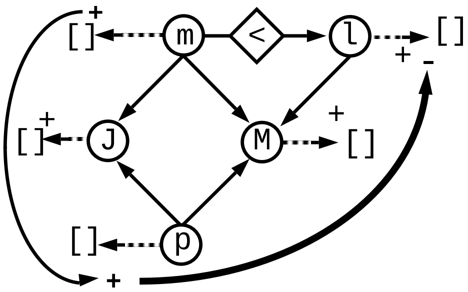

Figure 3: Graphic representation of the RM of (4).

The justification between the signs of tokens `m` and `p` is uncertain (graphically represented by a thin curved line), whereas the one that links the signs of `p` and `l` is certain (thick curved line). The reason for this distinction is that the meeting implies persuading in a very weak sense (it is a precondition), while persuading (to stay at home) entails non-leaving.

Note furthermore that in Figure 3 `l`'s attachment is labelled with two signs: the positive one (uncertain) models the presupposition of the leaving and the negative one (certain) reflects the fact that the leaving actually did not take place. The last sign is the *preferred* sign (and it overrides the uncertain one); graphically, this is represented putting it near the end of the arc.

Justifications are needed not only by abstract completeness considerations, but also to deal with counterfactual utterances, as is explained in the next section.

3.4 The implementation of *eval* and *int* functions

At this point, the structure of the RMs should be clear. Now, I present via a couple of examples the algorithms that implement the *eval* and *int* functions (that build an RM for an utterance and evaluate a question in an RM, respectively). Both algorithms can be defined in the same way (by structural recursion on the *logical form* of an utterance, see below), therefore I describe only the way the model of an utterance is built.

A raw RM is built on the ground of LK and ontology and is then refined using content knowl-

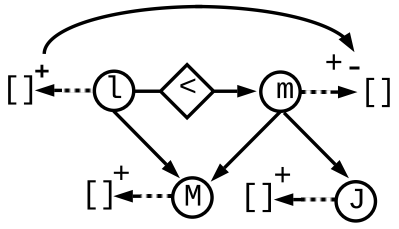

Figure 4: Graphic representation of the RM of (6).

edge. Let us consider for example the RM of utterance (6) represented graphically in Figure 4. The following steps take place during its creation:

- token `l`, from 'left', is created and it is attached to the space `[]` with a positive and certain sign. The sign is certain because of linguistic considerations: 'left' belongs to the main proposition;
- token `M` is created and it becomes the value of slot `agent` of token `l`. Now, the building of the RM of the main proposition is terminated;
- token `m`, from the event of the secondary proposition, is created and attached to `[]`.[13] The sign of this attachment is still positive, but uncertain because the event is in a secondary proposition;
- the slot `agent` of token `l` assumes as value the token `M`, already present in the RM; token `J` is instead created and it becomes the value of slot `theme` of token `m`;
- the temporal relation between the tokens `l` and `m` is created;
- all the above operations take place on the ground of linguistic and ontological considerations. However, to complete the construction of the RM, some content inferences are needed to create a negative certain sign (preferred to the positive uncertain one) on the attachment of `m` and the corresponding justification.

Thus, the division of linguistic, ontological and content work seems clear. Linguistically and ontologically, tokens are created, slot values are filled, relations explicitly referred in the utterance are produced and attachments are created. On the ground of content considerations, justification arcs, representing the causal relations between events implicit in the utterance, are added, and the same happens for new signs.

However, the separation between LK, ontology and content is not so simple: temporal relations may be created on the basis of content, and justifications on the basis of LK. This happens, for example, in the creation of the RM of (7), that is similar to the one represented in Figure 4: the only differences are the attachment of `m` (that is labelled by only one negative certain sign) and the justification (that is certain too). In this case, the temporal relation is created on the basis of the content, in that the fact that the leaving takes place before the meeting is indubitably a content inference. Furthermore, the justification derives from LK considerations, in that it appears explicitly in the word 'if' of the utterance.

As already specified, the discussion above regards exclusively the function *int*. Nevertheless, the algorithm that implements the function *eval* can work in a similar way; instead of creating tokens, it verifies that they already exist in the RM.

The algorithm implementing *eval* must work in a particular way for the evaluation of counterfactual utterances. Such evaluation takes place in three steps: first, the antecedent and the consequent of the counterfactual utterance are evaluated in the current RM; second, the current RM is modified accordingly to what it was said in the antecedent of the counterfactual, obtaining the counterfactual model; third, the consequent of the counterfactual is evaluated in the counterfactual model. Let us consider the evaluation of utterance (7) in the RM for (4) (the RM in Figure 3). The evaluation takes place in the following way:

- the antecedent and the consequent of (7) are evaluated in the RM; both of them are *false*

---

[13] The attentive reader might note that the processing of the clause containing the presupposition, the secondary one, takes place after the main one's. This is in contrast with the nature of the presuppositions, which should be tackled as first. But, I pointed out in footnote 2 in Section 2.1, here it is the 'a posteriori' aspect of temporal presuppositions (and of the whole sentences encompassing them) that is studied, so this is not a relevant difference.

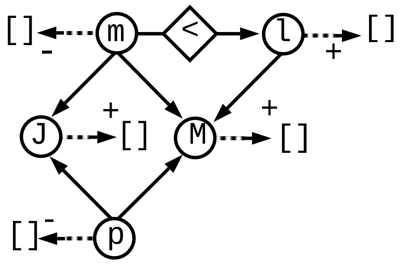

Figure 5: Counterfactual model for the evaluation of (7).

(and they must be *false* in order to evaluate the counterfactual utterance *true*);

- the counterfactual model, that is obtained modifying the original RM in such a way that the antecedent is evaluated *false*, is illustrated in Figure 5. Observe that the token `m` is attached with a negative sign to `[]`, in that the antecedent must be evaluated *false*. This, by means of the justification between the signs of `m` and `l` (see the original RM in Figure 3), leads to removing the positive sign on `p`'s attachment and labelling this token with an opposite (negative) one. The same happens with token `l`; here the removal of the negative sign brings up the positive sign;
- the consequent of the counterfactual ("Mary left") is evaluated in the counterfactual model, obtaining *true* as result. The counterfactual utterance itself is then evaluated *true*.

From the informal description in this section, it should not be difficult to extract the algorithms for *int* and *eval*, implemented in the system described in Section 4.

3.5 Related work

A brief look at related work is mandatory, in order to emphasize the differences between RMs and other proposals. In this section, researches on *discourse models* and *discourse representation theory* (DRT) are briefly compared with RMs, and it is shown how RMs can handle in a simple way the concepts of *belief* and *situation*.

RMs can be seen as models of previous discourse context, into which information from sentences is merged, and against which queries are evaluated. There are a lot of studies on *discourse models* in which it is investigated how the various structures that can be individuated in a discourse ought to be used to understand the meaning of the sentences forming such discourse: see for instance [22, 28, 31, 35, 37, 41]. RMs could be a new instrument for this research, even if it might be more appropriate to say that RMs are a computational tool for modeling the meaning of sentences, and that they do not seem to suffer from any intrinsic limitation for being used at the level of discourse.

RMs are also comparable to DRS (Discourse Representation Structures), the 'models' used in DRT [26], but here also there are some differences. First of all, Kamp and Reyle themselves say in their book on DRT [26, page 627] that they don't tackle the problems I have analyzed here:

> There exists the possibility of using before-phrases in a kind of "virtual" sense which is not possible for prepositional phrase with after. In a case where the sentence "George died before the completion of his novel" is true, the completion of the novel presumably never took place. [...] This use of before has given semanticists a good deal of trouble. [...] It is an issue which we will not pursue here.

Notwithstandig that, one might try to treat temporal presuppositions in DRT—and encounter some difficulties. Consider for instance the standard DRS of utterance (6) reported here

"Mary left before meeting John", (6)

namely the DRS of Table 2. The DRS is divided in 3 groups, separated by empty lines: the first one models the main clause, the second one the word 'before', and the third one the subordinate clause. In such DRS there is nothing representing the facts that the event e_2 (the meeting one) is only presupposed (and then uncertain), that it has not happened, that there is a causal link between the occurrence of the two events and there is no 'first-order' object representing the occurrence of the events.

$t_1\ n\ e_1\ x\ t_2\ e_2\ y$
$t_1 < n$
$e_1 \subseteq t_1$
$mary(x)$
$e_1 : leave(x)$
$t_1 < t_2$
$t_2 < n$
$e_2 \subseteq t_2$
$john(y)$
$e_2 :\ meet(x, y)$

Table 2: The DRS of (6).

Obviously, DRS could be extended in the direction indicated by RMs, but this is not so simple, in that in DRS there is nothing like RMs' spaces, attachments, signs and justifications, which are central concepts in RMs. So, DRSs might be situated at a semantic level, while RMs work on the semantic-pragmatic boundary: a DRS is more similar to a logical form [2] than to an RM.

RMs can be extended in a natural way for taking into account the concepts of *beliefs* and *propositional attitudes* [5, 15]. For instance, spaces allow to easily represent Mary's *intention* to leave in utterances (1) and (4): it is sufficient to attach the token `l` in Figures 2 and 3 to a space, `[int(M)]`, representing the world of the events that should have happened if everything had gone as presupposed. In this way, one can create a family of operators on worlds (`int(X)` for intentions, `bel(X)` for beliefs, and so on), indexed on the tokens of the RM. These operators can transform one world (for instance `[]`) in other ones (`[int(M)]`, `[bel(M)]`, etc.)

Finally, RMs might easily be improved for handling utterances like

> "Mary left with George. This hurt John"

in which it is not the event per se that 'hurt John', but the whole context. In order to treat this kind of utterances, it will be necessary to introduce the concept of *situation* [5, 15] in RMs: the RM of this utterance could look like the one in Figure 6, where the grey circle is the graphic representation of 'what hurt John'.

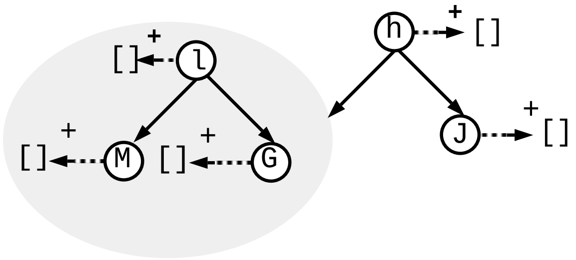

Figure 6: Situations in RMs.

4 The TOBI system

This section presents TOBI, a system that communicates with the user in natural language (English) and uses the RMs illustrated in the previous sections as internal representations of utterances. TOBI is implemented in LPA Prolog on a Macintosh, and it can handle all the examples presented in Section 2.1 (and similar ones). The following subsections illustrate: the class of natural language processing systems to which TOBI belongs, the architecture of the system, its data flow, and its internal data structures.

4.1 Comprehension systems

TOBI is a natural language processing system. It is indeed a particular case of such systems, a *comprehension system* (CS): it has the *unique* aim of interacting with the user in natural language. This section describes a CS using the concepts presented in Section 3.1 and, on the basis of this description, some design choices made in TOBI are motivated.

A CS simulates the typical human activities of comprehension and production of natural language utterances: it can understand a discourse (sequence of utterances) and provide correct answers to questions regarding the discourse. For the sake of simplicity, only *polar* questions are considered, i.e. questions admitting as answers only 'yes' (*true*), 'no' (*false*) or 'I don't know' (*unknown*). In Figure 7 an example of dialogue between a hypothetical CS and a user is shown.

```
User> Mary met John before she left.
User> Did Mary leave?
CS> Yes.
User> Did Mary kiss John?
CS> I don't know.
User> Ann met George before she left and
      he persuaded her to stay at home.
User> Did Ann leave?
CS> No.
```

Figure 7: An example of interaction CS - user.

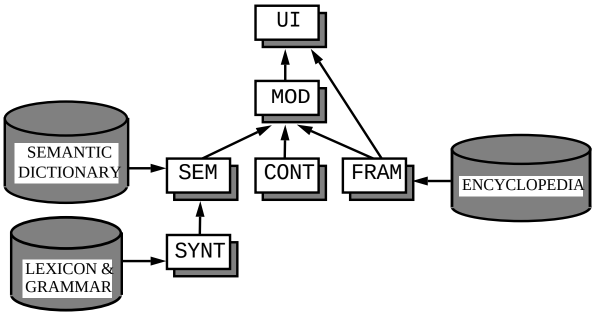

Figure 8: TOBI's architecture

CSs work by building some internal representation of a discourse, and using such representation to answer successive questions. On the basis of what it was presented in Section 3.1, the implementation of a CS can be accomplished in two ways. The first (and traditional, see [20]) one is to build a nonmonotonic inferential system, that uses an inference procedure $\vdash$ (and usually a TMS, Truth Maintenance System). This kind of CS will be named CS *Formulae & Inference* (F&I). The second way of realizing a CS is to implement a system that builds a (computable) model of the discourse and evaluates the question in that model in order to obtain the right answer. Systems of this kind are named CS *Models & Evaluation* (M&E), and (with respect to CS F&I) work at the more primitive and flexible level of models and model-based evaluation.

TOBI is a CS M&E that uses the above described RMs to model the meaning of utterances. Since CSs F&I may rely on well known basis, developed in mathematical logic, the attempt to follow the new way of CSs M&E must be justified. The most persuasive critique of CSs F&I concerns the way they have to abort the process of inference if they obtain no answer. This is an unnatural way of working, and it has no cognitive plausibility. On the other hand, CSs M&E present many interesting features: they seem to have more cognitive plausibility (it is widely recognized that human beings build a model of the utterance they hear, and that they don't use an inferential mechanism to answer questions), they might deal with the problem of termination in a better way than CSs F&I do, and they show a natural treatment of implications weaker than entailment (like the presuppositions met in the examples in Section 2.1).

These observations motivate the attempt to follow the approach of CSs M&E. However, it must be said that CSs F&I are preferable in handling entailments and incomplete knowledge, fields in which the inferential approach demonstrates all its power.

Summarizing, TOBI is a CS M&E, and not a F&I one for the following reasons:

- the kind of phenomena it has to deal with: mainly presuppositions, not entailments;
- the greater cognitive plausibility;
- the supposed better control of the weakening of the system's inferential capacities;
- the examination of what can be done using models and evaluation in place of classical and well known logical calculi.

4.2 TOBI's architecture

Figure 8 presents the architecture of TOBI. Here is a list of TOBI's modules with a short description of their tasks:

- `SYNT`: morphoSYNTactic analyzer that parses the input utterance, producing its *syntactic structure*. `SYNT` uses a *lexicon* and a *DCG grammar* [19] as knowledge bases;
- `SEM`: SEMantic analyzer; it takes the syntactic structure produced by `SYNT` and produces as output the *logical form*, that is a representation of the utterance in a slot-filler notation, in which events and semantic roles are

singled out. This module uses a *semantic dictionary* associating syntactic terms with the corresponding concepts;

- `FRAM`: FRAme Manager; manager of the *encyclopedia* (a taxonomy of categories and concepts) and models. It implements the procedures needed to work on classes (the encyclopedia) and instances (the models);
- `CONT`: the module devoted to handling CONTent knowledge;
- `MOD`: MODel builder; module that implements the functions *int* and *eval* using procedures from `SEM`, `FRAM` and `CONT`;
- `UI`: User Interface; it accepts utterances from the user (via keyboard) and answers his (her) questions. This interface is developed using the features of LPA Prolog for windows and menus management.

4.3 TOBI's data flow

In Figure 9 the data flow of TOBI is presented, in order to illustrate the process that takes place when the system builds an RM from an utterance. TOBI processes the utterance in three steps. The first step is the *morphosyntactic analysis*: the input utterance is parsed into its syntactic structure.

The syntactic structure is input to the *semantic analysis*, that produces another representation of the initial utterance, namely its logical form.

The last step is the *interpretation*: here the logical form is used to build the RM of the utterance (or, more generally, to *integrate* the old RM with the new information in the utterance). It is in this phase that TOBI's peculiarity comes in evidence. In most natural language systems, content knowledge is encapsulated in the encyclopedia, together with ontological knowledge. In TOBI the two kinds of knowledge are separated; the encyclopedia contains only ontological knowledge, that can easily be dealt with in symbolic terms; the content part is handled by another module.

As it was said in Section 2.2, the phenomenon of temporal presuppositions is based on the ontology of time, not on its content. But a system that works only at an ontological level could do very little. For example, to understand utterance (6) content considerations are necessary for keeping into account the relation between the leaving and meeting events. Then TOBI has to deal with content inferences too. I have assumed that ontology can be handled using classical symbolic methods; there are reasons, however, to believe that this might not be true for content (see for instance [1]). Furthermore, the linguistic phenomena studied rely on the ontology, not on the content. Therefore, in the present version of TOBI, content inferences are replaced by an interface to an external user, activated upon request of a master module, which fully implements ontological inferences. The clear division between ontology and content gives a conceptually clean system, and the implementation of a 'real' content module can be tackled in an independent way.

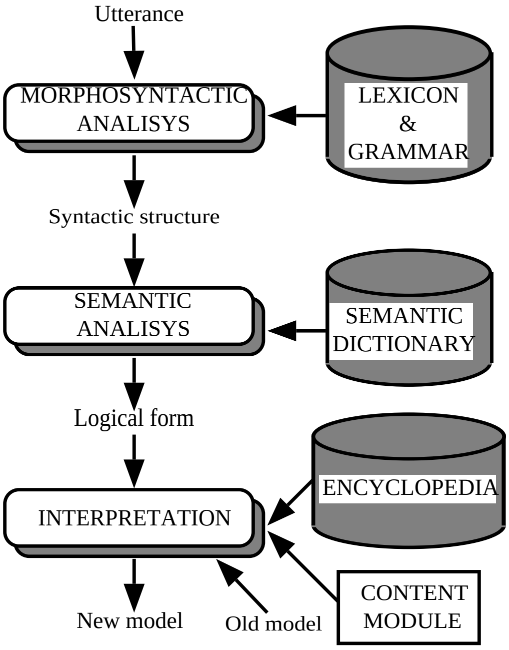

Figure 9: TOBI's data flow for the interpretation of an utterance.

4.4 TOBI's data structures

In this section I go deeply into the internal details of TOBI's work, illustrating in a concrete example the utterance analysis process. The data structures passed across the three steps described in Section 4.3, namely the utterance, the syntactic structure, the logical form and the RM, are

```
1 s(asser,
2    vg(sing,meet,trans,ind,past,aff),
3    subj(np(sing,f,det,[],
              pNoun(person(mary)),[])),
4    obj1(np(sing,m,det,[],
              pNoun(person(john)),[])),
5    obj2(nil),
6    [es(
       prep(before),
7      s(asser,
         vg(sing,leave,intr,ind,past,aff),
9        subj(np(sing,f,det,[],pronoun,[])),
10       obj1(nil),obj2(nil),[]))])
```

Figure 10: The syntactic structure that TOBI generates when interpreting (1).

explicitly shown. Let us consider the interpretation of utterance (1)

"Mary met John before she left" (1)

The syntactic structure, that the `SYNT` module builds starting from the utterance (1), is the Prolog term showed in Figure 10, where (see [42] or [2] for a description of the terminology used here):

- `s` stands for 'sentence' and `asser` means that the sentence is assertive;
- line 2 represents the verb group (`vg`) that is singular, has head 'meet', is transitive, is in the indicative form, in the past tense and affirmative;
- line 3 models the subject of the main clause; it is a noun phrase (`np`), singular, female, definite, without modifiers (`[]`), with head the proper noun 'Mary' and without qualifiers (`[]`);
- lines 4 and 5 represent the direct ('John') and indirect object (not present here) of the main clause, respectively;
- in lines 6 to 10, the embedded sentence (`es`) introduced by the temporal presupposition 'before' is represented, in a recursive manner. The symbols have the same meaning as in the main clause.

The syntactic structure is then input to the `SEM` module that (recursively) transforms it in the logical form of Figure 11. Here the events (meet and leave) and the semantic roles (agent and theme) are singled out and the anaphoric references are made explicit.[14] The notation should be clear, after noting that the logical form is expressed in a slot-filler notation, that `sLf` and `npLf` stand for 'sentence logical form' and 'noun phrase logical form' respectively and that `<VAR>` stands for an unspecified value.

The last data structure is the recursive model. The RM of (1) was illustrated in Figure 2. In TOBI, it is represented as the set of Prolog facts of Figure 12, where, again, the meaning should be clearly understandable, when compared with the graphic representation of Figure 2.

5 Conclusions and future work

The main points discussed in this paper are:

- the linguistic phenomena of temporal presuppositions and counterfactuals;
- the distinction between linguistic and extra-linguistic knowledge, and the role played by different kinds of knowledge and inferences (entailments, presuppositions, ontology and content) in the linguistic phenomena studied;
- the abstract syntax of the fragment of language related to temporal presuppositions and counterfactuals;
- the recursive models, an instance of computational models for naturally dealing with temporal presuppositions and counterfactuals. I have sketched the formal specifications of recursive models, described their structure and compared them with related proposals;
- the consideration that the nonmonotonic linguistic phenomena of temporal presuppositions and counterfactuals are more naturally handled by comprehension systems Models & Evaluation than Formulae & Inference;
- the implementation, based on the RMs, of the TOBI system, a comprehension systems Models & Evaluation indicating that RMs are an effective tool for treating temporal

---

[14] I know that this is not an easy problem, but here I am not interested in it. In TOBI, anaphoric references are handled via a simple *history list* mechanism (see [2]).

```
sLf(meetConc(aff),asser,past,
  [slot agent:
       npLf(person,sing,f,<VAR>,slot name:mary,slot sex:f),
  slot theme:
       npLf(person,sing,m,<VAR>,slot name:john,slot sex:m),
  slot atTime(before):
    sLf(leaveConc(aff),asser,past,
      slot agent:
       npLf(person,sing,f,det,slot name:mary,slot sex:f))])
```

Figure 11: The logical form of (1).

```
model(m1, inst(leaveConc1, leaveConc)).
model(m1, inst(person2, person)).
model(m1, inst(person1, person)).
model(m1, inst(meetConc1, meetConc)).
model(m1, instanceSlot(leaveConc1, agent, person1)).
model(m1, instanceSlot(meetConc1, theme, person2)).
model(m1, instanceSlot(person2, sex, m)).
model(m1, instanceSlot(person2, name, john)).
model(m1, instanceSlot(meetConc1, agent, person1)).
model(m1, instanceSlot(person1, sex, f)).
model(m1, instanceSlot(person1, name, mary)).
model(m1, relation(beforeTime, meetConc1, leaveConc1)).
model(m1, attach(a4, meetConc1, [])).
model(m1, attach(a3, leaveConc1, [])).
model(m1, attach(a2, person2, [])).
model(m1, attach(a1, person1, [])).
model(m1, attachSign(a4, (s4,plus,cert))).
model(m1, attachSign(a3, (s3,plus,uncert))).
model(m1, attachSign(a2, (s2,plus,uncert))).
model(m1, attachSign(a1, (s1,plus,uncert))).
```

Figure 12: TOBI's internal representation of the RM of Figure 2.

presuppositions and counterfactuals and that the dichotomy ontology-content seems reasonable.

From an epistemological point of view, RMs make explicit some considerations about the use of *negation* by human (or more generally living) beings (see [6, 10]). In fact, the first way that one can imagine for representing the non-existence of an object (or the non-occurrence of an event) is probably the use of a slot 'existence' ('occurrence'), with the opportune value for each token. In RMs, the more general mechanism of spaces, attachments and signs allows not only to deal with existence and occurrence, but also to explicitly represent the fact that the causal relations hold between occurrences (or non-occurrence) of events, and not merely between events.

In the near future, TOBI will probably be enhanced in various ways. To extend the set of cases it can deal with, an extension of the vocabulary is needed. This, in conjunction with an improvement of the grammar, will allow for the treatment of utterances syntactically different from the ones considered in this work, but with some common semantic-pragmatic characteristics. For example, counterfactual phenomena are very common in language, and do not need a specific syntactic construction: another common case is for instance the use of the verb 'to wish', as in "Mary really wishes she had left". Also, the extensions regarding beliefs and situations illustrated in Section 3.5 will surely be considered.

Finally, it is also planned to formalize the theory that underlies the RMs, on the basis of Allen's theory of action and time [3, 4], of McDermott's temporal logic [32], and of Fomichov's theory of K-calculuses and K-languages [17] using the formal specifications presented in Section 3.2.

Acknowledgements

I am indebted to Marco Colombetti for many valuable suggestions, and for long and stimulating discussions; also to Paolo Giangrandi, Elena Not, Fabio Rinaldi, and two anonymous referees, for many useful comments on an earlier draft of this paper.

References

[1] G. Airenti and M. Colombetti. Ontology of mind, subjective ontology, and the example of temporal presuppositions. Technical Report 92–018, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, March 1992.

[2] J. Allen. *Natural Language Understanding.* The Benjamin/Cummings Publishing Company, Inc., 2nd edition, 1994.

[3] J. F. Allen. Mantaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[4] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[5] J. Barwise and J. Perry. *Situations and attitudes.* MIT Press, Cambridge, MA, 1983.

[6] G. Bateson. *Steps to an Ecology of Mind.* Chandler Publishing Company, 1972.

[7] H. K. Berg, W. E. Boebert, W. R. Franta, and T. G. Moher. *Formal methods of program verification and specification.* Prentice Hall, 1982.

[8] D. Bochvar. On three-valued logical calculus and its application to the analysis of contradictions. *Matematiceskij Sbornik*, 4:353–369, 1939.

[9] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[10] M. D. S. Braine. On the relation between the natural logic of reasoning and standard logic. *Psychological Review*, 85(1):1–21, 1978.

[11] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense.* Cambridge University Press, Cambridge, 1991.

[12] C. C. Chang and H. J. Keisler. *Model Theory.* Studies in Logic. North Holland, Amsterdam, 1973.

[13] C. J. Date. *An Introduction to Database Systems*, volume 1. Addison-Wesley, Reading, MA, 5th edition, 1989.

[14] E. Davis. *Representations of Commonsense Knowledge.* Morgan Kaufmann, San Mateo, CA, 1990.

[15] K. Devlin. *Logic and information.* Cambridge University Press, 1991.

[16] H. B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York, 1972.

[17] V. A. Fomichov. A mathematical model for describing structured items of conceptual level. *Informatica*, 20(1):5–32, 1996.

[18] G. Gazdar. *Pragmatics: Implicature, presupposition and logical form.* Academic Press, 1979.

[19] G. Gazdar and C. S. Mellish. *Natural Language Processing in Prolog.* Addison Wesley, Wokingham, England, 1989.

[20] M. R. Genesereth and N. J. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann Publishers, Los Altos, California, 1987.

[21] M.L. Ginsberg, editor. *Readings in nonmonotonic reasoning*, Los Altos (California), 1987. Morgan Kaufmann.

[22] B. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July–September 1986.

[23] O. Heinämäki. Before. In J.N. Peranteau, P.M.and Levi and Phares G.C., editors, *Papers from the Eighth Regional Meeting of the Chicago Linguistic Society*, Chicago, Illinois, 1972. University of Chicago.

[24] O. Heinämäki. *Semantics of English Temporal Connectives.* PhD thesis, Department of Linguistics, University of Texas at Austin, Austin, Texas, 1974. Reproduced by Indiana University, Linguistic Club, November 1978.

[25] J. R. Hobbs and R. C. Moore, editors. *Formal Theories of the Commonsense World.* Ablex, Norwood, NJ, 1985.

[26] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publisher, Dordrecht, The Netherlands, 1993.

[27] L. Kartunnen. Presuppositions of coumpound sentences. *Linguistic Inquiry*, 4(2):169–193, 1973.

[28] A. Lascarides and N. Asher. Discourse relations and defeasible knowledge. In *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, pages 55–62. University of California at Berkeley, June 1991.

[29] A. Lascarides and J. Oberlander. Temporal connectives in a discourse context. In *Proceedings of the European ACL*, pages 260–268, 1993.

[30] S. C. Levinson. *Pragmatics.* Cambridge University Press, Cambridge, 1983.

[31] W. C. Mann and S. A. Thompson. Rhetorical structure theory: A theory of text organization. Reprint ISI/RS-87-190, Information Science Institutes, University of Southern California, 4676 Admiralty Way/Marina del Rey/California, June 1987.

[32] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6(2):101–155, 1982.

[33] S. Mizzaro. TOBI - An Ontological Based Interpreter for Temporal Presuppositions and Counterfactuals. In R. Trappl, editor, *Cybernetics and Systems '94*, volume 2, pages 1879–1886, Singapore, 1994. World Scientific. Proceedings of the Twelfth European Meeting on Cybernetics and System Research, organized by the Austrian Society for Cybernetics Studies, held at the University of Vienna, Austria, 5-8 April 1994.

[34] S. Mizzaro. Recursive models: A computational tool for the semantics of temporal presuppositions in natural language. In *Proceedings of the fifth meeting of the AI*IA (Italian Association for Artificial Intelligence), Workshop 'Natural Language Processing'*, pages 43–46, Naples, Italy, 26-28 September 1996.

[35] L. Polanyi. A formal model of the structure of discourse. *Journal of Pragmatics*, 12:601–638, 1988.

[36] H. Reichenbach. *Elements of Symbolic Logic.* Macmillan, London, 1947.

[37] R. Scha, B. Bruce, and L. Polanyi. Discourse understanding. Technical Report 391, Bolt Beranek and Newman Inc., 10 Moulton Street, Cambridge, MA, October 1986.

[38] P. Smolensky. On the proper treatment of connectionism. *Behavioral and Brain Science*, 11(1):1–74, 1988.

[39] I. Sommerville. *Software Engeneering.* Addison-Wesley, Reading, MA, 3rd edition, 1989.

[40] R. Turner. *Logics for artificial intelligence.* Ellis Horwood Limited, Chichester, England, 1984.

[41] B. L. Webber. Discourse deixis and discourse processing. Technical Report MS-CIS-88-75, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19014, September 1988.

[42] T. Winograd. *Language as a cognitive process*, volume 1: Syntax. Addison-Wesley, Reading, MA, 1983.