

From the textual description of an accident to its causes

Daniel Kayser*, Farid Nouioua¹

Laboratoire d'Informatique de Paris-Nord, UMR 7030 du C.N.R.S. – Institut Galilée, Université Paris 13, 99 avenue Jean-Baptiste Clément, F 93430 – Villetaneuse, France

ARTICLE INFO

Article history:

Received 29 September 2008

Received in revised form 7 April 2009

Accepted 23 April 2009

Available online 6 May 2009

Keywords:

Natural language understanding

Causal reasoning

Norms

Inference-based semantics

Semi-normal defaults

ABSTRACT

Every human being, reading a short report concerning a road accident, gets an idea of its causes. The work reported here attempts to enable a computer to do the same, i.e. to determine the causes of an event from a textual description of it. It relies heavily on the notion of *norm* for two reasons:

- The notion of cause has often been debated but remains poorly understood: we postulate that what people tend to take as the cause of an abnormal event, like an accident, is the fact that a specific *norm* has been violated.
- Natural Language Processing has given a prominent place to deduction, and for what concerns Semantics, to truth-based inference. However, *norm*-based inference is a much more powerful technique to get the conclusions that human readers derive from a text.

The paper describes a complete chain of treatments, from the text to the determination of the cause. The focus is set on what is called “linguistic” and “semantico-pragmatic” reasoning. The former extracts so-called “semantic literals” from the result of the parse, and the latter reduces the description of the accident to a small number of “kernel literals” which are sufficient to determine its cause. Both of them use a non-monotonic reasoning system, viz. LPARSE and SMODELS.

Several issues concerning the representation of modalities and time are discussed and illustrated by examples taken from a corpus of reports obtained from an insurance company.

© 2009 Elsevier B.V. All rights reserved.

1. Motivation

1.1. Basic postulates

The work described here is grounded on two postulates:

- what is perceived as the cause of an event is:
 - the norm itself, if the event is perceived as normal,
 - and the violation of some norm, if the event is considered abnormal;
- the semantics of natural language (NL) is not based on the notion of truth, but on norms.

* Corresponding author.

E-mail addresses: Daniel.Kayser@lipn.univ-paris13.fr (D. Kayser), Farid.Nouioua@lipn.univ-paris13.fr, Farid.nouioua@univ-cezanne.fr (F. Nouioua).

¹ Now at Laboratoire des Sciences de l'Information et des Systèmes, UMR 6168 du C.N.R.S. Université Paul Cézanne (Aix-Marseille 3), Avenue Escadrille Normandie-Niemen 13397 Marseille Cedex 20, France.

The notion of norm, which plays a central role in this paper, has both a descriptive and a prescriptive meaning.² In the *descriptive* sense, norms are just what explains the difference between what is perceived as normal or not. In the *prescriptive* sense, norms build a corpus on the basis of which an agent is considered, legally or otherwise, entitled or not to perform an action.

In Artificial Intelligence, these two meanings have given rise to two rather separate fields of study. On the one hand, non-monotonic logics have been developed in order to derive conclusions that are considered *normal*, in the absence of any specific circumstance invalidating this derivation. On the other hand, deontic logics have the purpose of formalizing the reasoning of agents respecting *normative* prescriptions [8,17,48].

In this paper, norms will generally be taken in their descriptive sense but clearly, as it is normal to follow the rules, this acceptance of norms includes as a special case the prescriptive sense, and we will also have to deal with duties, which are normative.

Our first postulate concerns a very old and very controversial issue: when is it sensible to say that A causes B? Determining the essence of the notion of cause is not in the agenda of Artificial Intelligence. However, commonsense reasoning makes an intensive use of causation, e.g. for diagnosing, planning, predicting; and therefore AI cannot (and does not, see e.g. [33,55]) completely ignore the debate concerning this notion. What AI needs, however, does not concern the metaphysics of cause, but only how people reason causally. And, even if observation reveals that we use the word *cause* to mean rather different things, i.e. that this word is polysemic, in a vast majority of cases we take as causal for an abnormal event the fact that some agent has violated a norm. We report in the paper a psychological experiment showing which violation(s) are selectively chosen as cause(s) of the event.

Consider now our second postulate: dealing with a sentence such as:

The car before me braked suddenly.

A truth-based approach (see e.g. [13,27,35]) will derive all the conclusions C that logically follow from the existence of a time *t* and a car *c*, such that the two following propositions are true at *t*: (i) *c* brakes suddenly and (ii) *c* is located in front of the writer. Clearly, several other propositions, none of them being valid in the logical sense, come to the mind of a human reader, e.g. (iii) at *t*, the writer was driving, (iv) s/he was driving in the same direction as *c*, (v) no vehicle was between *c* and him/her, (vi) s/he had to act quickly, in order to prevent an accident, and so on. Subsequent information may force the reader to retract some of these conclusions. Nonetheless, as they are likely to be present in the mind of every person who shares our culture, there is no necessity to consider separately the propositions C derived by means of truth-preserving operations (the only ones that are said to pertain to semantics, according to the prevailing theories), from the propositions derived by means of norms (generally said to be the result of pragmatics).

Knowing the norms of a domain is absolutely necessary to understand the texts of that domain. But there exists no exhaustive list of the norms ruling any given domain: the rules and regulations are only the visible part of the iceberg of all we know, and keep implicit, about the domain. An indirect consequence of our study is to point out that examining how people ascribe causation to the events happening in a domain is a powerful means to reveal its implicit norms.

1.2. Specification of the goal

In order to validate our postulates, we need to focus on a domain where the number of norms remains reasonable, where abnormal events are frequent, where these events are reported in natural language, and where it is easy to ask people what they take as being the cause of the events reported.

We selected the domain of road accidents, for the following reasons:

- A large number of short texts exists, describing such accidents: every insurance company receives daily a number of forms filled by drivers, describing what happened to them.
- Most people of our culture know enough about car crashes to give sensible answers, after having read a text, when they are asked what caused the accident.
- Each report describes some facts, but clearly implies also a number of other facts, which are not logically entailed. We can therefore see whether our postulates work, i.e. check whether a reasoning based on norms captures the kind of reasoning used by the reader.
- An accident by itself is an anomaly, and the text generally goes back to another anomaly that allegedly explains why it happened. We can thus test which anomaly, if any, is taken to be the cause of the accident, and confirm or infirm our first postulate.
- The number of norms involved is neither too large nor too small. They are clearly not limited to those listed in the Highway Code.
- The corpus on which we perform our study has been studied from different points of views. (See for example [60,19,34]. The last work presents a system that produces automatically a 3D scene of an accident from its textual description.³)

² Von Wright [62, Chap. 1] discusses in much greater details the various meanings of the word *norm*.

³ Another study [63] concerns British records of road accidents: the authors use a description logic in order to check the consistency between a natural language report and coded data which are both components of the record.

However, a drawback of this corpus is worth a mention. Most of the reports sent by a driver to an insurance company are, for obvious reasons, biased in order to lessen the author's responsibility; the rhetorical aspect of a plea interferes with the more basic descriptive part of the report, be it truthful or not. The fact that the texts do not necessarily reflect what really happened is not *per se* a problem: the norms of a domain are not better revealed from “true” descriptions, whatever this may mean, than from biased ones. The unwanted consequence of the choice of this corpus is that it requires some effort to identify (and most of the time, to ignore) what has been added for pure argumentative reasons. A study is currently in progress that examines specifically the argumentative strategies used by the authors to highlight or to minimize some aspects of the accident, in order to convey the best possible impression of their behavior [9]. As we will see, this dimension of the reports does not affect much the results of the present work.

We obtained, by courtesy of MAIF, an insurance company, a number of reports written in French. Some of them are unclear, and only the accompanying drawings make them understandable. We discarded them and kept only those which are self-contained, i.e. those on the basis of which we understood enough of what happened to build a hypothesis that we could justify about the cause of the accident.

Our first intention was to write a program that finds the same cause(s) as human readers. This objective presupposes that human readers agree on that cause(s). But a recent experiment, conducted with colleagues from the Laboratoire Travail et Cognition of Université Toulouse-Le-Mirail, on reports that appeared to us as having a very clear cause, shows that subjects have a wider diversity of opinion than what we expected. Therefore, to start with, we want to obtain – and we actually obtain – a good correspondence between our intuition of the cause and the result of our program. This proves that we have a reasonable model of our own causal reasoning. Moreover when the subjects of the experiment give an answer differing from our expectation, they seem to give a higher importance than we do to some norms, but not to use something else than the notion of norm to ascribe the cause, and this shows that we have captured to a fair extent some general features of human causal reasoning.

1.3. Methodology

The strategy of development could seem simpler if we started from the language side and proceeded through different steps of the architecture in the natural order (morpho-syntax, semantics and pragmatics). However, this would require solving many hard linguistic problems without knowing whether they are or not crucial for the task. It turns out that much of the linguistic complexity (scope of modals, precise interpretation of the grammatical tenses) is relevant only for the argumentative part of the report (e.g. *at first, I thought that ... , then I realized that ...*), which will be dropped in the core of the search for the causes of the accident.

We adopted a reverse strategy: we started with a corpus of 73 texts, later called the *training sample*, on the basis of which we identified “semantic literals” i.e. a set of concepts that are relevant for causal reasoning. This set must satisfy two main conditions:

- To be accessible from the texts, i.e., to represent pieces of information evoked explicitly in the texts.
- To be sufficiently fine-grained to discriminate between two texts sharing many similar features, but for which the cause of the accident, according to human readers, is different.

The method we used to identify this set ensures the respect of the above conditions; we took in turn the different sentences in the texts of the training sample, and we extracted manually from them the concepts they refer to, following two intuitive principles: the first one is to ignore the passages written for purely argumentative reasons, as well as the description of the damage when it is of no help to determine the cause of the accident. The second principle is to stay at a level of granularity very close to that of the text: we choose concepts that are a more or less the direct translation of words or linguistic expressions; whenever this is possible without altering the causal ascription, we merge into a single concept the linguistic terms that differ only by their connotative power. We defined this way around 50 “semantic literals” that seem necessary and sufficient to carry the endeavor to a successful end.

Moreover, we soon identified a limited subset of semantic literals called the “kernel”, in terms of which all possible causes can be expressed (see Section 6.1.1).

After having chosen the set of semantic literals from the training texts, the next step of our strategy has been to develop the semantico-pragmatic reasoning⁴ on these texts, under the assumption that we would somehow succeed in extracting the semantic literals from the texts. The linguistic reasoning was later developed analogously, i.e., going back up for each text, from its required result, the semantic literals, to the input necessary to get this result: this input consists in a set of so called “linguistic literals”, i.e. syntactic relations between the words of the text, and we assumed that we would get them from a parser.

While developing the semantico-pragmatic and the linguistic reasoning, we continued to collect reports, keeping them aside for validation purposes. As appears clearly from the results (see Section 8), the selection of the semantic literals and

⁴ The semantico-pragmatic reasoning infers the cause of an accident described in a text from its semantic literals.

of the kernel was fairly good, since it allows to analyze correctly most of the new reports, although many of them differ considerably from those of the training sample.

Finally, as the parsers available did not work to our satisfaction, we developed our own parser and added a set of post-treatment heuristics to adapt its output to the required input of the linguistic reasoning.

The semantico-pragmatic and the linguistic reasoning represent the main parts of the system, more importance being given to the former, because it implements our hypotheses about norms and causes. In contrast, the parsing task had not the same importance with respect to the objectives of this work; it was developed to ensure a complete chain of treatment, at a late stage of the project, when all the reports were already available. That is why we did not consider it in the validation process.

1.4. Plan of the paper

The plan of the paper is as follows:

Section 2 introduces the notion of causation as seen from an AI point of view.

Section 3 gives an overview of the general architecture of the system. All its modules do not have the same importance: our study focuses on the semantico-pragmatic processing, which corresponds to the causal reasoning. The morpho-lexical and syntactic treatments are needed to build a complete chain from the text to the cause, but they have been treated more cursorily, and admittedly contain several *ad hoc* features; they contain some original points, however and are therefore briefly presented.

Section 4 presents the representation language used throughout of the paper, or rather the different layers of representations that are needed to progressively elaborate the cause from the textual input. The general idea is to use a first-order reified language that simulates some second-order features, like quantifying over properties and also some kind of modal reasoning. The formulas of this language represent the norms as non-monotonic rules.

Section 5 describes the linguistic reasoner, i.e. the process that takes place once the parser has completed its work: it uses commonsense knowledge to correct some mistakes made by the parser, solves anaphora, infers semantic propositions and assigns each proposition to a temporal state.

Section 6 presents the semantico-pragmatic reasoner, i.e. the process that starts from the semantic literals obtained from the linguistic reasoner and uses common knowledge to infer implicit information about the situation described in the text. This reasoning process stops as soon as the cause of the accident is determined and expressed as the violation of a norm.

In Section 7, we discuss some implementation issues. We first give a brief overview of the “answer set programming” paradigm. Then we explain how we use it, i.e. how we translate the default inference rules into extended logic programs.

Section 8 is devoted to the validation. It has two subsections:

- One in which we assess the agreement between the causes found by the system and what we considered intuitively to be the cause of the accident when we selected the text in our corpus. Unsurprisingly, the agreement is very good on the 73 training texts, but it is still remarkable for the 87 texts kept for validation. Moreover, in order to test the robustness of the main modules of the system, we have considered two inputs; their actual input, and an ‘ideal’ input crafted manually, in order to contrast their results when they are operated with the noisy data received from the upstream modules, with their performances when they get ‘clean’ data.
- A second in which we confront our results with the various answers provided by the subjects of the psychological experiment on 10 texts (see Appendix A) selected for their representativeness. We then present our conclusions.

2. Causation and AI

As noted in the introduction, causation is an extremely controversial notion. AI has no need to take side in the controversy. But as a user of the notion, it can notice that, at least in the most usual cases, what humans take as “the” cause(s) of an event is/are the *most abnormal* factor(s) that enable the event to take place. Here, the word factor is taken to be absolutely neutral: a cause can be a physical or a mental event, a non-event i.e. the fact that some expected event did not occur, a general or a specific principle or law, or whatever comes to mind when we are asked what caused something to happen.

A revealing example, taken from one of the most influential books on causation [45], goes as follows: an explosion happens after someone lit a cigarette; what is the cause of the explosion? If the event takes place at home, the answer could be: “*the gas was leaking*”; if it occurs in an oil refinery, it is sensible to answer: “*smoking in such places is forbidden*”. In the first case, smoking is not abnormal, but the presence of flammable material is; the opposite is true in the second case.

The extremely rich literature on causation dates back to antiquity and seems, even nowadays, still deeply influenced by controversies of bygone ages. Anyway, it is of little help to get what we want, namely some hints to discover the cause of an abnormal event described by a text. There are indeed many studies that aim to find out explicit causal links in texts (e.g. for French [21]), to discuss the difference in meaning between the diverse means that language uses to express causal links [52] or to extract automatically causal knowledge from NL texts [12]. But this does not tell how to answer “why”-questions [40], when the author did not make explicit the causal links in the text.

A large number of psychological experiments have been or are being conducted to examine how subjects attribute causal power, under which circumstances they accept causal transitivity, how causation relates to explanation, what parameters affect judgments of causal responsibility (see e.g. [31,29]).

Most of the time, we are interested in knowing the cause of x only as long as we can *do* something to make x happen if we want it, or to prevent it from happening, if we do not want it. This is the reason why, among the various theories concerning the nature of causation, the most usable ones for everyday life are the ones based on *intervention*. Admittedly, the interventionist conception does not fit with some very common causal assertions, such as: *the moon causes the tide*, except if we consider that these sentences evoke a metaphorical situation where an intervention would remove the moon, and where we could observe that there is no longer tide.

But for most current uses of the notion of cause, it is easy to bring to light one or several *actions* that justify resorting to that notion. Many of the technical approaches to causal reasoning make a more or less explicit appeal to this idea: propagation in [15], determination of exogenous variables in [33] are ways by which the idea of acting and tracing the effect of the action, are brought into the reasoning. Even philosophers opposed to an interventionist conception of causality (e.g. [37]), contrast an allegedly ‘objective’ notion of *cause* with a more flexible notion of *causal responsibility*.

Therefore, the most appropriate AI tools to reason on causes are closely related to the representation of actions, and the most influential AI works about causation incorporate theories of action. For instance, Drew McDermott’s [47] predicates *pcause* and *ecause* are defined in the framework of so-called chronicles, i.e. dense sets of states ordered in a tree-like time structure; a node in the tree represents the choice between doing or not doing an action (the representation framework of [5] can be seen as an elaboration of this early work). The non-monotonic causal logic discussed in [25,1] represents causality by introducing a new causal operator to define causal rules. The resulting causal theories are used to represent properties of actions like concurrently executed actions, non-deterministic actions, or actions with conditional effects. Another formalization of the causal dependencies between actions and their direct and indirect effects is proposed in [24]. In this work, a conditional logic is used where the conditional implication is interpreted as a causal implication. In every case, the notion of action goes with the notion of conditions enabling the action to succeed, i.e. the well-known qualification problem (cf. e.g. [23]), and with the propagation of its effects, i.e. the ramification issues (cf. e.g. [61]).

Pearl’s [55] fundamental work on causality assigns a central role to an operator, *do*, in order to make explicit the difference between what follows from an observed correlation and from an intervention. [30] introduces the notion of “*actual cause*”, i.e. the cause that explains effectively a given event in a specific scenario. This idea, which somehow matches our concern of “primary anomaly” (see below Sections 4.3 and 6.5), is however developed in a quite different context; it originated to meet the needs of control theory where, the equations describing a physical system being given, the problem is to determine the variables on which one could act to obtain desired behaviors or avoid undesired ones. Halpern and Pearl use structural equations to capture the dependencies of the so-called endogenous variables (the ones on which one can act) on other endogenous variables and on exogenous variables (external to the system). The interventionist aspect of their theory is clear from the fact that they take the cause of an event as being the modification (that can only result from an intervention) of the value of endogenous variables, but not of exogenous ones.

In our context, we do not have a model of a physical system, nor do we want any: as a matter of fact, even if our texts describe a physical world, they should not be represented according to scientific physics. The choice of scientific physics would not only be computationally expensive, since it would include a large number of parameters; but it would be also representationally inadequate: the value of a number of parameters is neither present in, nor derivable from, the text; even worse, the texts are not written by experts in mechanics, but by drivers likely to share the misconceptions about force and energy that are common in the population [32]. What makes sense for them may not be translatable at all in terms of scientific physics. Finally, the cause perceived by the readers is generally not amenable, contrary to control theory, to a change in the value of a parameter.

Dubois and Prade [18] tackle a problem that is more similar to ours, as they are interested in perceived causality, and do not suppose that the domain is represented as a set of structural equations. They use instead Kraus, Lehmann and Magidor’s relation of non-monotonic deductibility [38,39]. Despite the fact that this relation has not been developed to take into consideration a temporal evolution, they use it to express the beliefs of agents concerning the normal course of events: an agent who believes that in the context C , B follows normally from A accepts the deductibility relation $C \wedge A \sim B$. If in that context, s/he observes $\neg B$ at time t , takes $\neg B$ to be persistent, but observes that event A occurs at time t and that B becomes true at time $t + 1$, then s/he will take A as being the cause of B .⁵ Our approach differs from theirs by at least two aspects:

- when we express a norm, we find necessary to make explicit its temporal direction: in a given situation, we can have both normal expectations concerning the future, and normal postdictions, and they should not be confused with one another.
- [38] is a non-monotonic scheme that does not acknowledge explicitly the possibility of having several extensions. However, the multiplicity of extensions corresponds to a real phenomenon in natural languages: some texts are truly

⁵ A comparison between several approaches to causal ascription, precisely in the domain of road accidents, can be found in [4].

ambiguous. Even if this phenomenon is seldom present in our corpus, we prefer to use tools that are able to account for it, hence our choice for another representation language (see Section 4).

3. Architecture of a Natural-Language understanding system

3.1. General

AI researchers have always sought to build systems that understand NL. After the first limited, but spectacular, attempts to answer questions asked in NL on a narrow domain [26] or to solve problems expressed in NL [7], they turned to more difficult problems and adopted a stratified architecture: morpho-lexical issues are solved first, then comes a parser; the result of the parser is interpreted semantically in order to build a representation of the literal meaning, and finally a pragmatic step finds how this meaning makes sense in the interaction between a speaker/writer and a hearer/reader. Each step is carried out with as much care, and as many resources as possible, in order to feed the next step with the most reliable data.

However, despite the remarkable increase in the capabilities of computers, this architecture has had rather limited success, for at least two reasons: the goal of each separate step can be defined at a theoretical level, but when faced to concrete texts, it is often unclear what the 'correct' solution is, and the agreement between annotators can be reached only through artificial conventions. The second reason is that the requirement of optimality at each step is misguided: we make perfect sense of texts full of spelling, syntactic, and/or semantic mistakes, as long as we have enough information from other sources to compensate for the corrupted items. It seems that language comprehension is more the result of an equilibrium between a number of linguistic and extra-linguistic constraints, than the result of a one-way deductive process starting from a textual input and yielding an end product, whatever its nature.

The only reason why the stratified architecture is still the most widely used is that the alternatives are much more difficult to construct: architecture based on blackboards, semantically-driven parses, multi-agent systems, and so forth have been tried, but for lack of principled rules, they have generally worked only in restricted environments.

The architecture of the system that we developed for the present study is not very original: it looks more or less like the stratified ones, and suffers from several ad-hoc features. Its original features are nonetheless as follows:

- its basic mechanism is inference; each stage works under the hypothesis that its input is imperfect, and that its results can be imperfect too; therefore, the logic underpinning all inferences is non-monotonic;
- there is no separation between semantics and pragmatics;
- it is goal-oriented, i.e. all the architecture is designed in order to converge towards one among a small variety of possibilities. Each stage can thus be seen as a reduction: there are fewer dictionary entries than there are morphological forms, less concepts than words, and far less potential causes of an accident than there are combinations of relevant concepts.

This architecture is therefore obviously inadequate to handle arbitrary texts for arbitrary purposes, but we believe that there is more to learn in factoring out what makes various task-oriented systems work satisfactorily, than from conceiving a general language toolbox.

More specifically, the architecture of our system is described on Fig. 1. Before examining in turn its various components, we give an idea of the corpus on which they operate.

3.2. The corpus

As said earlier, we started with 73 reports, our training sample, and kept aside the reports that we collected afterwards. All reports have been filtered with the same criterion, i.e. we discarded those that we did not consider as self-contained. We ended up with 160 reports, among which we selected for their diversity a sample of 10 reports, to be used in the psychological experiment. This sample is provided in Appendix A and the full corpus is available at the web page <http://www.lipn.univ-paris13.fr/~kayser/corpus160.pdf>. The spelling mistakes and the most blatant stylistic inadequacies have been corrected, the names of people and of locations have been erased, but all the rest, including lexical oddities and discrepancies with academic French, has been kept.

The following figures give an idea of the nature of the corpus: the 160 reports contain 380 sentences altogether. 58 reports (36%) have only one sentence; one report has 10 sentences. The length of a report goes from 7 to 184 words. The length of a sentence goes from 2 to 48 words. The overall size of the sample is 6790 words, but only 1242 distinct forms occur, and these forms stem from 868 distinct dictionary entries (the maximum number of forms per dictionary entry is 14, and it is reached for the two auxiliaries *être* (to be) and *avoir* (to have)).

3.3. Morpho-lexical level

In some sense, we can say that we skipped most of the difficult issues at this level: we have provided our system with the full list of forms appearing in the corpus; each form is annotated with as many syntactic classes as it takes in the

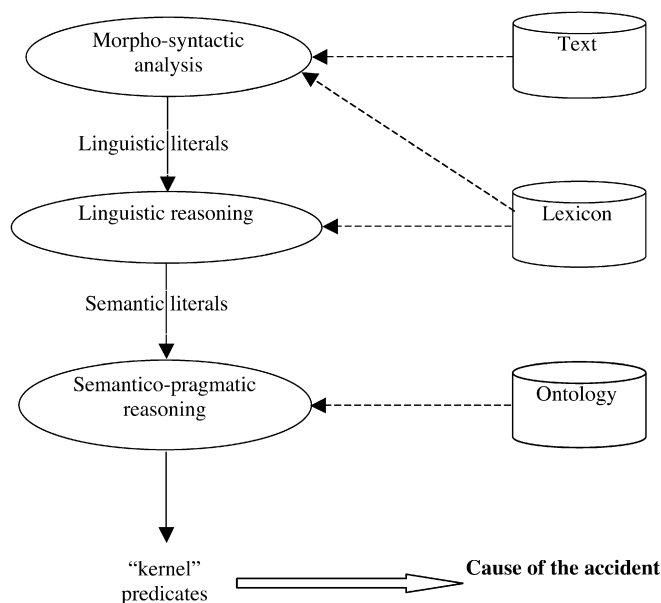


Fig. 1. Overall architecture of the system.

corpus. We have identified 10 main syntactic classes: adverbials (A), subordinations (B), coordinations (C), determiners (D), nouns (N), pronouns (O), prepositions (P), qualifiers (Q), relatives (R), and verbs (V), with subclasses (gender and number for D, N, O and Q; tense and person for V). For example, we have:

avancée VsFS
 avant NMS
 avant QNS
 avant P

i.e. *avancée* (which can be a noun in French) is only used in the corpus as the verb (V) *avancer* (to move forward) at the tense past participle (s) feminine (F) singular (S). On the contrary *avant* has three possible syntactic classes that are all present in the corpus:

- a noun (N) e.g. *l'avant du véhicule* = the front of the vehicle; in that case it is masculine singular (MS),
- a qualifier (Q) e.g. *la porte avant, le phare avant* = the front door, the front light; in that case, it is singular but can be used both for masculine and for feminine, hence the N for neutral,
- a preposition (P) e.g. *avant le choc* = before the shock.

The only difficulties we solved are elision (E) and polylexical units (i.e. units extending over several words). The entry of each elided form is provided with its full form and for each polylexical unit, we selected as the 'key' to the unit, either its first (L) or its last (S) word, depending on which one is less likely to be used autonomously. We put with the key the list of the other words of the unit. For instance

aux E-à-les
 afin L-deP
 conséquent S-parA

means respectively that:

aux is the elided form of the sequence *à les*,
afin is the first word of the polylexical unit *afin de* (so as to), which is a preposition (P),
conséquent is the last word of the polylexical unit *par conséquent* (consequently), which is an adverbial (A).

3.4. Syntactical level

The multiplicity of syntactic classes, as well as the ambiguity between some polylexical units and the same sequence built with isolated words, can lead, even for short sentences, to hundreds or thousands of distinct decompositions. Only a couple of them are worth considering for a parse. We have adopted a solution similar to the old systèmes-Q [14]: instead of

first building all the possibilities, then eliminating those which are impossible, we have considered a directed acyclic graph; each choice corresponds to a node having several successors, but the different alternatives have the same successor. Once the graph is built, a small number of rules are applied in order to prune all the paths that would contain a sequence of categories that cannot be found in French, or at least are completely implausible in car crash reports. This technique reduces the number of paths to be considered to at most a few dozen.

The next step is to group sequences of categories which build together noun phrases, verb phrases, and prepositional phrases. It is easy to replace in the graph these sequences by a macro-unit. Then, the sequence of macro-units is examined in order to find which ones follow regular patterns of French sentences. Again, only a few paths survive among those remaining.

Each path is then analyzed, and the syntactic relations between its elements are made explicit.

Example. *Je circulais sur la voie de droite.* (lit. I drove on the lane of right.)

| | |
|--------------------------------------|---------------------------------------|
| subject (circulais, je) ⁶ | subject (drove, I) |
| compl_v (sur, circulais, voie) | verbal complement (on, drove, lane) |
| compl_n (de, voie, droite) | noun complement (of, lane, right) |
| qualif_v (circulais, imparfait) | verb qualification (drove, imperfect) |

The results obtained through this technique contain some mistakes. The reasons are that (i) we focused on the most common constructions, neglecting correct but less ordinary turns that are nonetheless present in the corpus, and (ii) the writers sometimes ignore the rules of academic language, and assume rightly that the reader will find the correct meaning out of an incorrect sentence. Rather than writing more complex rules, we prefer to yield erroneous predicates, and rely on the subsequent stages to put things right.

3.5. Discussion

We could (and actually we did) use morpho-lexical and syntactic analyzers available off the shelf. Several preliminary experiments dissuaded us to do so. One of the best morpho-lexical analyzers for French, “Tree-tagger⁷”, yields generally good results on our sentences, but makes a number of blunders that are easily avoided, as soon as the domain is known. For example, it always takes *chaussée* (causeway) to be the past participle of the verb *chausser* (to put a shoe on); it is disturbed by expressions like *véhicule A*, *véhicule B* which are extremely frequent in our corpus, and randomly assigns categories to *A* and *B*. We tried to write ad hoc rules to correct these blunders but, after a while, we found more practical to do the work by ourselves (managing one thousand forms remains rather easy; if we had to handle a vocabulary bigger, say, by an order of magnitude, the adaptation of Tree-tagger, or of a similar program, would be a better choice).

Similarly, we have tried to use commercially available or free parsers for French. The resulting trees were almost unusable. Our first move has been to write a grammar which encompasses as exactly as possible the specificity of our corpus, and to get a complete analysis of each sentence. We have therefore written a set of around 400 context-free rules. But some of them were extremely ad hoc and, even after several improvements, the number of parses of a sentence, while low on average, could climb to nearly 1000 in the worst cases.

When we realized that the subsequent steps did not need the complete structure of a sentence, but only some easily available relationships, and moreover that erroneous relations were not as harmful as we thought, we opted for the solution described above.

4. Representation

The different steps of our treatment are designed in order to converge towards the identification of the cause of an accident. We first identified a few predicates (the *kernel*) that are sufficient to identify what, according to our intuition, is the cause of every accident reported in our corpus (Section 6.1.1). As transforming the result of the parse into a set of propositions using only the predicates of the kernel cannot be done in a single step, we implemented a sequence of translations. Each step of the sequence consists in inferring conclusions using ‘new’ predicates, presumably closer to the kernel, from premises expressed with ‘old’ predicates, closer to natural language. We could thus say that each of the treatments represented in Fig. 1 is an inference engine working on a different layer of a representation. However, all layers share a general format.

4.1. Reification

The different layers are first-order languages. They differ in their set of predicate symbols: the result of the parse uses grammatical predicates, such as *subject*, *qualification*; the semantico-pragmatic reasoner uses predicates such as *same-lane*,

⁶ In order to distinguish between several occurrences of the same word, the arguments of the syntactic relations are the ranks of the words, instead of the words themselves; we show here the words, for readability.

⁷ Tree-tagger is available at the web address: <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>.

follows; the kernel contains predicates such as *has-control*, *is-stopped*. They differ in their arguments: the result of the parse has words, later treatments have concepts and temporal states. However, in order to homogenize as much as possible the various treatments, we follow a few general principles.

The first principle, rather wide-spread in AI, is reification. It prevents resorting to higher-order logics and, in some sense, it allows a representation to cope with modalities without introducing modal logics. As a matter of fact, writing:

```
holds (follows, veh_A, veh_B)
```

instead of:

```
follows (veh_A, veh_B)
```

makes it possible to stay in first order logic, while factoring out every property that is common to, say, positional predicates. A terminological difficulty arises here: *follows* is normally a predicate symbol, but technically it will be used as a constant. In the rest of the paper, we reserve the term *predicate* for predicates of the reified logic (e.g. *holds* is a predicate) while we use *property* to denote what would have been a predicate in ordinary logic (e.g. *follows* is a property); therefore properties will encompass very heterogeneous concepts, like actions, effects, and so on.

We immediately meet a problem: *holds*, like every predicate, must be given a fixed arity, whereas the statements to be represented may involve a variable number of arguments. We circumvent it by adopting the following convention:

holds is a ternary predicate; its arguments are a property in the above sense, an agent, and a temporal marker.

Whenever the property involves other arguments, a binary function *combine* creates a new property from the property name and these extra arguments. For instance, to express that vehicle A follows vehicle B at time 3, we write:

```
holds (combine(follows, veh_B), veh_A, 3)
```

i.e. there exists a complex property, namely “*to follow veh_B*”, that the agent *veh_A* satisfies at time 3. Should the arity of the basic property be greater than two, the application of *combine* would be iterated as many times as necessary. This convention is analogous to the treatment adopted in Montague grammars [51]: the statements (type *t*) are obtained by the application of elements of type (*t/e*) to elements of the universe (type *e*; here, the agents), where the elements of type (*t/e*) can either be intransitive verbs, or they can be built from elements of type ((*t/e*)/*e*) (i.e. transitive verbs) combined with an argument of type *e*; similarly, *combine* creates a unary property from a binary one and an argument.

Another drawback of reified languages is that every boolean operator on properties must be axiomatized, instead of being given for free in the standard languages. However, it turns out that we do not need explicit operators on properties other than the negation. Therefore we introduce a function *not* and we have the axiom:

$$(1) (\forall P) \text{ holds } (\text{not}(P), A, T) \leftrightarrow \neg \text{holds } (P, A, T)$$

4.2. Modalities

One of the benefits of reification is the fact that, in reified logic, the predicates play the role of the modalities in ordinary logic; so we can use as many of them as we need, without introducing new symbols.

Duties play an important role in our application: most of what is perceived as abnormal can be traced back to an agent who does not comply with his/her duties. To put it another way, this corresponds to an interventionist conception of causality: we are interested in the causes of an event *e* insofar we can find someone that can be held as responsible, not necessarily in a legal sense, but in the following one: an agent endowed with free will, had at some time in the past the possibility of doing or not doing an action, and according to his/her decision, the normal course of events that followed includes or not the event *e* [36]. So the fact that an agent had the duty of doing/not doing an action and did the opposite is a good candidate for being the cause of an anomaly. We write:

```
duty (P, A, T)
```

to denote the fact that at time *T*, agent *A* has the duty of making *P* true, where *P* can be any simple or complex property (e.g. *not(P')* or *combine(P', A')*).

But in many circumstances, agents may have duties that they are not able to comply with; for instance an agent has the duty of stopping but has for some reason lost control over his/her vehicle. In that case, the cause of the accident is rather the reason of the loss of control, than the fact that the agent did not stop when s/he should have done so. To express this kind of situation, we introduce another modality:

```
is_able_to (P, A, T)
```

to denote the fact that at time *T*, agent *A* has the possibility of making *P* true, where again *P* is a simple or complex property.

The two above modalities look like a necessity and a possibility, and in a sense, they are respectively so. But the accessibility relation with regard to which they are so, is not the same. Therefore, we do not have the usual duality: it is obvious that not having the duty to stop is quite different than having the capacity to move.

Another modality, the availability of an action will be presented in due time (Section 6.4).

4.3. Anomalies

Our goal is to identify the cause of an accident, or rather its cause as it is understood from a written report which may be truthful or not. Our postulate says that, as an accident is perceived as abnormal, the cause must be an anomaly. We therefore need to have in our language a predicate denoting abnormalities. The previous section alludes to the fact that two kinds of anomalies exist:

- an anomaly is derived when it can be explained by another earlier anomaly; an agent who does not stop when s/he must stop, has an abnormal behavior; however, if this anomaly gets an explanation in the text (e.g. the agent lost control because of an icy patch), the anomaly is a derived one;
- an anomaly is primary whenever nothing in, or implied by, the text allows to attribute it to an earlier event.

Therefore our language contains two distinct predicates:

```
Derived_anomaly (P, A, T)
Primary_anomaly (P, A, T)
```

to denote that something abnormal regarding property P happened for agent A at time T , and that this anomaly is respectively a derived or a primary one.

4.4. Non-monotonicity

According to our views, understanding natural language amounts to reasoning with both linguistic and extra-linguistic premises. The notion of norm, which plays a central role, goes with the idea of exception: a norm describes what happens generally, but not always. The logics incorporating the notion of exception are non-monotonic.

In practice, we need non-monotonic logics to cope with situations where further information changes our account of the cause of an accident. For example, if we read “... the driver of vehicle B lost control and hit my car ...”, the causal reasoning will find the loss of control as the primary anomaly, hence as the cause of the accident. Now, if the next sentence reads: “Vehicle B lost control because of an icy patch”, the causal reasoning will conclude that the loss of control represents rather a derived anomaly, and the primary anomaly will be the presence of the icy patch on the road.

AI has developed a large number of non-monotonic schemes, some of them being multi-extensional. We require multi-extensional logics for reasons of adequacy: as, in some cases, a sentence can yield several readings, the reasoning must have the capability to yield several sets of conclusions. But as we selected the texts of our corpus for their clarity, we expect to get a unique extension for each of them. As a matter of fact, getting several extensions has served as a signal for bugs in our system. Should nonetheless the case arise, then we would re-run the search for anomalies in each extension, and possibly find different causes of the same accident, depending on the reading.

The reasons for which some linguistically consistent readings have to be eliminated are heterogeneous: the polysemy of some words or of some grammatical constructions can be resolved with some knowledge of the domain. The pragmatics of writing to an insurance company plays a role too: if two interpretations I_1 and I_2 are linguistically possible, but I_1 would increase the responsibility of the writer while I_2 is neutral or favorable, it is extremely likely that I_1 is not the meaning intended by the writer.

All these reasons show that default conclusions which are acceptable to some point can be defeated when other considerations are taken into account, or in other words, that some default have priority over other defaults. Technically, this amounts to selecting a logic that is not semi-monotonic [11].

We have chosen Reiter's semi-normal defaults [56,57] because it is a non-monotonic, non-semi-monotonic logic endowed with multi-extensionality, and it is easily translatable into Answer Set Programs (see Section 7). Semi-normal default theories have extensions, except in very unlikely cases [20].

We adopt a convention that we find more readable than Reiter's original notation⁸:

Normal defaults will be written $A : B$ instead of: $\frac{A:B}{B}$.

Semi-normal defaults will be written $A : B [C]$ instead of: $\frac{A:B \wedge C}{B}$.

⁸ According to Reiter's convention, a default containing free variables stands for the set of defaults in which the variables are replaced by the elements of Herbrand's base. Moreover, the free variables of regular first-order formulas are considered implicitly universally quantified.

For implementation reasons, we limit the forms of inference rules. Let $A_1, \dots, A_n, B, C_1, \dots, C_k$ be first order literals. An inference rule in our system has one of the forms (i)–(iii):

$$\begin{array}{ll} A_1 \wedge \dots \wedge A_n \rightarrow B & \text{(i)} \\ A_1 \wedge \dots \wedge A_n : B & \text{(ii)} \\ A_1 \wedge \dots \wedge A_n : B [C_1, \dots, C_k] & \text{(iii)} \end{array}$$

(i) is the usual material implication: B is inferred in an extension whenever $A_1 \dots$ and A_n belong to that extension. (ii) is a “normal” default: B is inferred in an extension if $A_1 \dots$ and A_n belong to that extension as long as this is consistent. (iii) is a “semi-normal” default: B is inferred in an extension if $A_1 \dots$ and A_n belong to that extension as long as this is consistent and none of the $\neg C_i$ ($i \in [1, k]$) belongs to that extension.

4.5. Representing time

The reports describe a continuously evolving world. Should we represent time by means of continuous functions? This would be closer to physical reality and would allow us to use the scientific definitions of speed, acceleration, and so on. But as we explained in Section 2, what makes sense for the writers of the reports may not be translatable at all in terms of scientific physics.

A second difficulty concerns which time has to be represented. There are actually three times:

- the linear sequence of the actual events described in the report,
- the possible evolutions that the agents had in mind at different moments (e.g. *I thought he was going to turn left*),
- the time of writing (e.g. *I realize now that things happened that way, but on the spot I believed that they happened differently*).

It turns out that the “second time” is mostly used by writers to justify their behavior, showing that they did everything to achieve desirable goals and to avoid undesirable ones. We must not forget that the reports are not only – and not even mainly – descriptive: their goal is to reduce the responsibility of the writer in the accident. By limiting them to the representation of actual events, we miss much of their substance. On the other hand, creating a (sequence of) state(s) for every future that one of the protagonists is likely to have envisioned (like e.g. McDermott’s chronicles [47]) would increase, without necessity in our case, the number of issues to solve.

This is the reason why we postulated the following hypothesis:

Hypothesis. The detection of anomalies requires only the representation of states-of-affairs presented as having really occurred; the potential or counterfactual states play a role to determine the expectations of the protagonists, their reasons for acting or not acting, but need not be represented as states *per se*: their existence is implicitly assumed by the modalities that hold at states belonging to the actual unfolding of the event.

Corollary. *The temporal states can be represented by natural numbers.*

This hypothesis has given satisfactory results, while keeping the number of states very small (no more than 7 in the texts that we have analyzed).

4.6. Representing causation

We do not have, as in many other studies (e.g. [42,16,25]), a special connector reflecting causal implications. Instead of it, as will appear later, we have a predicate `pot_cause (Act, Eff)` that represents the common *belief* that an effect Eff is often the result of action Act , i.e. when reading a text reporting Eff , people have naturally in mind the possibility of Act . This predicate is not bijective: several actions might come to mind for a given effect, and reciprocally, an action can evoke several possible outcomes. By action, we mean rather the decision of an agent having the free will of performing or not performing it; so Act may be in fact $\text{not}(Act')$, i.e. the decision of not acting.

As in our view, causal reasoning is focused on an intervention, we *prefer* to ascribe as a cause of an abnormal event (the accident) a decision that is already known to provoke the observed effect. If this decision has been taken in an abnormal setting, e.g. by an agent who had a duty, who had the capacity of complying with this duty, and who decided not to comply, we assume to have found the primary anomaly.

Only, if this preference is not satisfied, i.e. if we are unable to find an intervention (action or failure to act) that explains the accident, we look for external causes. As the readers are willing to accept exogenous circumstances as causes in those situations, we must also accept non-agentive causes. This is the reason why we also have a property:

`Disruptive_factor`

that can be combined to other arguments (e.g. *icy patch*) to account for cases that are better explained without the intervention of an agent.

5. The linguistic reasoner

5.1. Post-treatments

The result provided by the syntactic analyzer is not completely suitable to be fed into the reasoning process. The first step consists thus, by means of three post-treatments, to process the “rough” product of the analyzer. The post-treatments are based on simple principles that take advantage of the specificity of the domain. Let us discuss them briefly:

5.1.1. Correction of erroneous subject/verb relations

As mentioned in Section 3.4 above, the parser makes a number of mistakes that have been left uncorrected. These mistakes concern mainly the predicates expressing subject/verb relations (8% of these relations produced by the analyzer are erroneous). We want first to detect these erroneous relations, and then to correct them by determining the correct subject for each verb.

We have used for that purpose a simple heuristic based on the compatibility between semantic features associated to a subset of the nouns and the verbs belonging to the lexicon. This information, which is very specific to the domain, has been coded into a boolean matrix. Whenever the relation *subject*(*V*, *N*) is detected as erroneous, the determination of the correct subject *N'* is based on a simple strategy, which gives acceptable results for our texts (see Section 8 for the results). We start searching the correct subject *N'* in the part of the text situated before the verb *V*, and only if this search fails, we look for *N'* after the verb *V*. During the first search (before *V*), if *V* is a present participle, or if it is followed by the relative “*qui*”, then it is most likely that the correct subject is not far from its verb and we look for the first compatible noun or pronoun before *V*. In the other cases (the most frequent ones), we start by looking for a noun *N'* which is already known as the subject of some verb; if it is semantically compatible with *V* (which ensures a kind of syntactic parallelism), we substitute *subject*(*V*, *N'*) for *subject*(*V*, *N*). Only if no such subject exists, we limit ourselves to the first compatible noun or pronoun. In all cases, if no solution has been found by searching before *V*, we look for the first compatible noun after *V*.

Example. Consider the following sentence (taken from Report B2.A of our corpus):

Report B2.A. *Le motocycliste qui circulait sur la file de droite d'une double voie à sens unique, a perdu le contrôle de son véhicule et a dérapé sur une flaque d'huile...*

(The motorcyclist who drove on the right-hand file of a double-lane one-way road, lost the control of his vehicle and skidded on a pool of oil ...)

The subject/verb relations provided by the parser are:

| | |
|-----------------------------------|------------------------------|
| subject (circulait, motocycliste) | subject(drove, motorcyclist) |
| subject (perdu, voie) | subject(lost, road) |
| subject (dérapé, contrôle) | subject(skidded, control) |

The first one is correct, the other two are not: neither “road” is the subject of “lost”, nor “control” the subject of “skidded”. The mistakes are detected: the noun “road” has the nominal feature *Lane*, which is not compatible with the verbal feature *Agentive-Verb* associated to the verb “to lose”. Analogously, the feature *Vehicle-State*, which characterizes the noun “control”, is incompatible with the feature *Move-Verb* associated to the verb “to skid”.

Once they have been detected, the mistakes are corrected; for the first relation, we look for the first subject situated before the verb “to lose” and compatible with it. As a result we find the noun “motorcyclist”.⁹ We apply the same strategy to the second relation and we find the same subject, as it has the feature *Person*, which is compatible with the feature *Move-Verb*.

5.1.2. “Supporting” verbs

The corpus contains a number of sentences, the main verb of which does not play the usual role of a verb, i.e. to predicate something of its subject; it merely introduces, and sometimes alters the meaning of, another verb. Such verbs are called, in [28], “supporting verbs”.¹⁰

For example, the sentence:

“il est venu heurter mon véhicule”
(lit. he came hit my vehicle)

⁹ The compatibility relation takes into account the frequent metonymic use of names of vehicles for their drivers and vice versa.

¹⁰ Actually, we use “supporting verbs” in a more general sense than the one defined in [28] and in other works of the linguistic literature. But as the general idea is analogous, we kept that term.

uses *venir* (to come) as the supporting verb for *heurter* (to hit). In this example, the “support” can be suppressed without altering much of the meaning of the sentence: as a matter of fact, we draw roughly the same conclusions from the above sentence and from:

“il a heurté mon véhicule”
(he hit my vehicle)

In this case, and in most of the other cases, treating the “supporting” verb boils down to replacing all its occurrences in the literals output by the analyzer, by the “supported” verb. Here the parser provides the relation `subject (venir, il)` (subject (come, he)); we simply replace it by `subject (heurter, il)` (subject (hit, he)). However, we treat as “supporting” verbs some cases which require further treatments. For instance, in the sentence:

“il a oublié de freiner”
(he forgot to brake).

the verb *oublier* (to forget), mainly present for argumentative reasons (“he” is a bad guy: he forgot to act when it was necessary), is treated as a supporting verb requiring the addition of a literal `qualif_v (freiner, neg)` (verb qualification (brake, neg)) to express that the subject/verb relation between he and brake has to be taken negatively. We have listed all the verbs of our corpus that play a role of support, and we have specified and implemented the corresponding treatments.

5.1.3. Anaphora resolution

The resolution of anaphora is renowned as a difficult problem in NL Processing. Our goal here is not to propose an original approach that could solve it in general. We simply describe a heuristic that adapts the well-known semantic consistency criterion [50] to our specific domain. We present here a sketch of our algorithm; further details can be found in [53].

We are interested in the determination of references only insofar the referents are active agents in the accident described (vehicles and persons). Accordingly, the set of possible referents contains the following agent names: *author* (the writer of the report), *veh_A*, *veh_B*, ..., *vehicle_name#1*, ..., *person_name#1*, ...¹¹ (as the texts have been anonymized, every detail which could identify the people involved in the accident has been replaced by generic names). For each text, we determine the list of expressions (nouns, pronouns and possessive adjectives) that are potential designators of an agent. We associate to each referent a property called its “nature”, which has initially the default value *vehicle* and which can be better specified during the execution of the algorithm by taking values like *car*, *motorbike*, *bicycle*, *truck*, ... We notice that the nature *vehicle* is compatible with all these terms while the terms are incompatible with each other.

In some cases, the reference is obvious. Words like *je*, *mon*, *ma* (I, my, me)... clearly refer to the writer and thus correspond to the reference “*author*”; similarly, symbols like: *A*, *B*... that are used to individualize vehicles, indicate in our system the agents: *veh_A*, *veh_B*, ... After having resolved the trivial cases, the remaining referents are considered one by one, according to their order in the text. To each unsolved referent, the anaphora resolution algorithm decides either to attribute the same reference as an already resolved referent, or to create a new reference. Moreover, after the resolution of a referent, some information is propagated which may help to systematically solve other referents (the *reference propagation*), or at least to refine the nature of the reference (the *nature propagation*) (see [53]). The idea behind the propagation is that if two words *X* and *Y* are connected by some specific syntactic relations, then they are constrained to refer to the same reference.¹² Hence, if one of the two words, say *X*, is a resolved anaphor, then the algorithm propagates its reference to *Y*. The same holds with *nature*: if the nature of one of the words, say *X*, is more specific than the nature of the other *Y*, then, the nature of *X* is propagated to *Y* and to all words having the same reference as *Y*. For example, in the expression “*ma moto*” (my motorcycle), the reference of “*ma*” is “*author*” (trivial resolution) and its nature is the default: *vehicle*; but as the nature of “*moto*” is *Motorcycle*, this nature propagates and the nature of “*author*” takes the value *Motorcycle* as well.

The decision to create a new reference is taken in two cases: the first case arises in particular linguistic contexts that reinforce the likelihood of a new reference. For example if the referent is a noun introduced by an indefinite article, or followed by a relative like *qui*, *que* (who, which), if it is the subject of a present participle, if it is the first referent introduced in the text or if it belongs to a set of nouns like *voisin* (neighbor), *adversaire* (opponent),... it is most likely that the noun creates a new reference. The second case is when no antecedent has a nature compatible with the current referent.

In contrast, if the set of compatible antecedents contains more than one element, the algorithm chooses the nearest reference while giving less priority to the reference “*author*”, because at this stage, it is unlikely that a referent indicating the writer is still not resolved.

¹¹ If in a given text, the same referent can simultaneously be designated by different expressions, we use the order provided in this section to select the name by which it will be referred to. But as the order is not critical, this choice does not really matter.

¹² If we have the relation `compl_n (de, X, Y)` and the reference of *Y* is *R*, then the reference of *X* is also *R*. For example, in “*le véhicule de nom_de_personne*” (Mr. So-and-so's vehicle) the reference of vehicle will be Mr. So-and-so. Similarly, if we have the relation `qualif_poss (X, Y)` and *Y* is a possessive, the reference of which is *R*, then the reference of *X* is also *R*. For instance, in “*ma moto*” (my motorcycle), “*ma*” has the trivial reference *author*, so the propagation assigns the reference *author* to *moto* as well. This rule has a few exceptions, e.g. in “my neighbor”, neighbor does not have *Author* as its reference! Clearly, these heuristics work only in a specific context, like ours.

Table 1

Application of the anaphora resolution algorithm on Report B2.A.

| Word | Reference | Nature | Type | Comments |
|--|-------------------------|-------------------|-------------|--|
| <i>Moto</i> ₁ Motorbike ₁ | <i>New</i> ₁ | <i>Motorcycle</i> | New | The noun “ <i>motocycliste</i> ” (motorcyclist) is the first referent introduced in the text |
| <i>Il</i> he | <i>New</i> ₁ | <i>Motorcycle</i> | Old | Only one antecedent: <i>new</i> ₁ . Its nature: <i>Motorcycle</i> is compatible with the nature of the referent: <i>Vehicle</i> , which then receives the more specific value: <i>Motorcycle</i> |
| <i>Moto</i> ₂ Motorbike ₂ | <i>New</i> ₁ | <i>Motorcycle</i> | Old | Only one antecedent: <i>new</i> ₁ . It shares the same nature: <i>Motorcycle</i> with the referent |
| <i>Mon</i> My | <i>Author</i> | <i>Car</i> | Trivial | Introduced at the beginning of the algorithm |
| <i>Véhicule</i> Vehicle | <i>Author</i> | <i>Car</i> | Propagation | Propagation by means of the relation: <i>qualif_n</i> (véhicule, mon) (<i>qualify_pos</i> (vehicle, my)) |
| <i>Moto</i> ₃ Motorbike ₃ | <i>New</i> ₁ | <i>Motorcycle</i> | Old | Two antecedents: <i>new</i> ₁ and <i>author</i> . By giving less priority to the author, we choose: <i>new</i> ₁ |
| <i>Voiture</i> Car | <i>Author</i> | <i>Car</i> | Old | Two antecedents: <i>new</i> ₁ and <i>author</i> . The nature <i>Motorcycle</i> of <i>new</i> ₁ is incompatible with the nature <i>Car</i> of the referent. We choose then the reference <i>author</i> . <i>mon</i> (my) and <i>véhicule</i> (vehicle) receive the nature <i>Car</i> by propagation |

Example (continued). Report B2.A of the corpus reads:

Report B2.A. “Le *motocycliste* qui circulait sur la file de droite d’une double voie à sens unique a perdu le contrôle de sa *moto*₁ et a dérapé sur une flaque d’huile. Il a été éjecté. Sa *moto*₂ est venue heurter *mon véhicule* roulant à son niveau sur la voie de gauche. La *moto*₃ a heurté l’avant, buté sur le côté droit de la *voiture*...”

(lit. The motorcyclist who drove on the right file of a double-lane one-way road, lost the control of his motorbike and skidded on a pool of oil. He was thrown out. His motorbike came to bump on my vehicle which was rolling at his level on the left lane. The motorbike bumped on the front, banged against the right side of the vehicle ...)

The first step of the anaphora resolution algorithm solves the trivial references and performs a first propagation that associates the reference *author* to the referents *mon* (my) and *véhicule* (vehicle). Table 1 shows how the algorithm computes the references corresponding to the remaining unsolved referents.¹³

5.2. Steps of the linguistic reasoning

The syntactic relations extracted from the text by the parser and adapted by the post-treatment are, together with the lexical information provided by the lexicon, the only explicit textual knowledge available to our reasoning system. This knowledge provides the set of premises of the so-called “linguistic reasoning”. This reasoning aims at constructing and expressing the explicit content of the text, in terms of the logical language described in Section 4. The literals that result from the linguistic reasoning have the format *holds* (P, A, T) and are called *semantic literals*.

Following a goal-oriented strategy, we have determined a limited list of 52 semantic literals (see Appendix B) that we estimate as being sufficient to express the relevant information for the next stages of the causal reasoning.

The role of the linguistic reasoning is thus, on the one hand to reduce the variety of the linguistic expressions present in our texts to a limited set of “semantic” concepts, and on the other hand, to find the temporal order of the events expressed by these concepts.

Let us notice that the limitation of the domain, as well as the use of a goal-oriented development methodology allow us to study only a limited set of linguistic phenomena, and to simplify their treatment. For example, we can choose safely and easily the appropriate senses of some words that are in general polysemic, and consider some words synonyms, even if this is not absolutely true in general.

The convergence process towards the semantic literals is performed gradually by the linguistic reasoning through three main steps. Each one uses inference rules of a different nature (see Fig. 2). Let us now explain the principles guiding each step.

5.3. Intermediate semantic literals

The first step of the linguistic reasoning consists in inferring a set of “intermediate semantic literals” from the subset of the linguistic literals based on the relations *subject/verb*, *verb/object* and *verb/complement*. The general form of an intermedi-

¹³ The new references are designated by *New*₁, *New*₂, ... At the end of the algorithm, these symbols are replaced by names of agents that are not yet used to designate other referents (see the predetermined list of these names earlier in this section). The column “Reference type” indicates the way by which the reference is calculated: it is either *trivial*, or obtained through the *propagation* mechanism, or *old* (when an antecedent already found is compatible); otherwise, it is *new*.

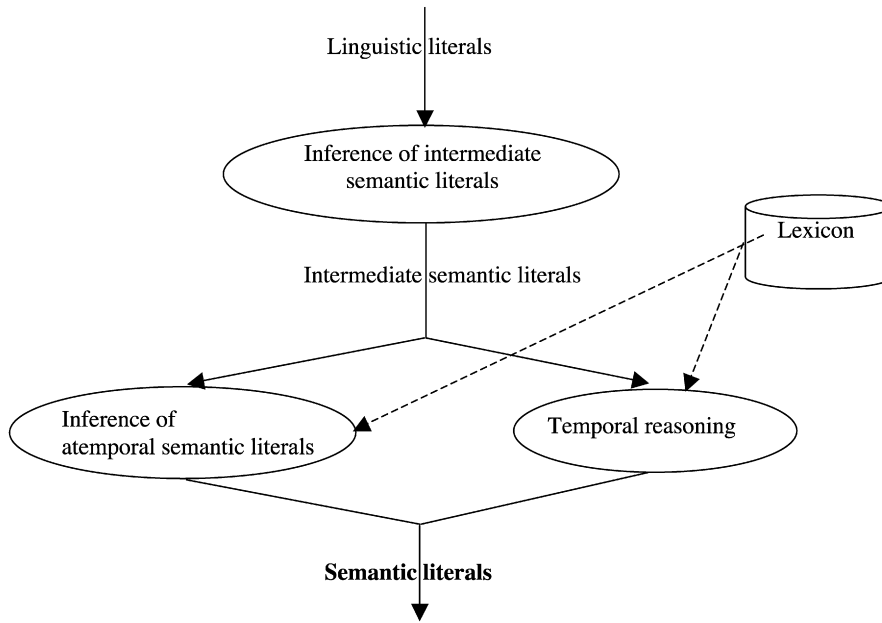


Fig. 2. Architecture of the linguistic reasoning process.

ate semantic predicate is:

`holds_I (P_Prop, P_Ag, temp_ref(W))`

The `_I` of `holds_I` stands for “intermediary” and distinguishes these literals from the final product of the linguistic reasoner. `P_Prop` and `P_Ag` stand respectively for “potential property” and “potential agent”. Intermediary predicates express a first supposition, to be confirmed or disconfirmed later on, about what might be the property and the agent of a genuine semantic literal. The parameter `temp_ref(W)` means that the temporal reference to be used as the temporal parameter in the semantic literal is provisionally related to the word `W` of the text.

This first step approaches the form of the semantic predicates, while remaining at a level which manipulates exclusively the words of the language rather than the concepts of the “real” world. This step is motivated mainly by a methodological reason: it replaces a subset of syntactic relations by more abstract predicates that are also more appropriate as premises for further inference.

The intermediate semantic literals are inferred by means of the five inference rules (2)–(6) given hereafter. Rule (2) is a normal default adapted to intransitive verbs. It infers the intermediate literals `holds_I (V, N, temp_ref(V))` from the relation `subject (V, N)`. Negative propositions are treated by rule (3), which inhibits the default (2). Analogously, rules (4) and (5) deal with transitive verbs and lead to literals of the form `holds_I (combine(V, O), N, temp_ref(V))` from the relations `subject (V, N)` and `object (V, O)`. Finally rule (6) is used to capture a relation between the verb, its subject and a prepositional phrase.

(2) `subject (V, N) : holds_I (V, N, temp_ref(V))`

(3) `subject (V, N) ∧ qualif_v (V, neg) → ¬holds_I (V, N, temp_ref(V))`

(4) `subject (V, N) ∧ object (V, O) : holds_I (combine(V, O), N, temp_ref(V))`
`[¬qualif_v(V, neg)]`

(5) `subject (V, N) ∧ object (V, O) ∧ qualif_v (V, neg) → ¬holds_I (combine(V, O), N, temp_ref(V))`

(6) `subject (V, N), compl_v (Z, V, C) : holds_I (combine(combine(Z, V), C), N, temp_ref(Z))`

Example.

Report B2.B.¹⁴ *Je circulais₁ sur₁ la voie₁ de₁ droite. Dans le virage, la moto a dérapé sur₂ des graviers. Je suis tombé de₂ l'engin qui a fini sa course sur₃ la voie₂ de₃ gauche. Le véhicule₁ A, circulant₂ sur₄ cette voie₃, n'a pu stopper et a percuté mon véhicule₂.*

I drove₁ on₁ the right lane₁. In the curve, the motorbike skidded on₂ bits of gravel. I fell from the machine, and it ended its way on₃ the left lane₂. Vehicle₁ A, who was driving₂ on₄ that lane₃, could not stop and hit my vehicle₂.

The set of linguistic literals extracted from the text and post-treated is the following:

| | |
|--|--|
| subject(circuler ₁ , author) | subject(drive ₁ , author) |
| subject(déraper, author) | subject(skid, author) |
| subject(tomber, author) | subject(fall, author) |
| subject(finir, author) | subject(end, author) |
| subject(circuler ₂ , veh_A) | subject(drive ₂ , veh_A) |
| subject(stopper, veh_A) | subject(stop, veh_A) |
| subject(percuter, veh_A) | subject(hit, veh_A) |
| object(finir, course) | object(end, way) |
| object(percuter, author) | object(hit, author) |
| compl_v(sur ₁ , circuler ₁ , voie ₁) | compl_v(on ₁ , drive ₁ , lane ₁) |
| compl_v(dans, dérapé, virage) | compl_v(in, skid, curve) |
| compl_v(sur ₂ , dérapé, gravier) | compl_v(on ₂ , skid, gravel) |
| compl_v(de ₂ , tomber, author) | compl_v(from, fall, author) |
| compl_v(sur ₃ , finir, voie ₂) | compl_v(on ₃ , end, lane ₂) |
| compl_v(sur ₄ , circuler ₂ , voie ₃) | compl_v(on ₄ , drive ₂ , lane ₃) |
| compl_n(de ₁ , voie ₁ , droite) | compl_n(∅, lane ₁ , right) |
| compl_n(de ₃ , voie ₂ , gauche) | compl_n(∅, lane ₂ , left) |
| qualif_n(véhicule ₁ , A) | qualif_n(vehicle ₁ , A) |
| qualif_poss(course, sa) | qualif_poss(way, its) |
| qualif_poss(véhicule ₂ , author) | qualif_poss(vehicle ₂ , author) |
| qualif_v(circuler ₁ , imparfait) | qualif_v(drive ₁ , imperfect) |
| qualif_v(déraper, passe_compose) | qualif_v(skid, present_perfect) |
| qualif_v(tomber, passe_compose) | qualif_v(fall, present_perfect) |
| qualif_v(finir, passe_compose) | qualif_v(end, present_perfect) |
| qualif_v(circulant ₂ , participe_present) | qualif_v(drive ₂ , present_participle) |
| qualif_v(stopper, passe_compose) | qualif_v(stop, present_perfect) |
| qualif_v(percuter, passe_compose) | qualif_v(hit, present_perfect) |

The application of rules (2) and (3) yields the following intermediate semantic literals:

| | |
|---|---|
| holds_I(circuler ₁ , author, temp_ref(circuler ₁)) | holds_I(drive ₁ , author, temp_ref(drive ₁)) |
| holds_I(déraper, author, temp_ref(déraper)) | holds_I(skid, author, temp_ref(skid)) |
| holds_I(tomber, author, temp_ref(tomber)) | holds_I(fall, author, temp_ref(fall)) |
| holds_I(finir, author, temp_ref(finir)) | holds_I(end, author, temp_ref(end)) |
| holds_I(circuler ₂ , veh_A, temp_ref(circuler ₂)) | holds_I(drive ₂ , veh_A, temp_ref(drive ₂)) |
| holds_I(percuter, veh_A, temp_ref(percuter)) | holds_I(hit, veh_A, temp_ref(hit)) |
| holds_I(stopper, veh_A, temp_ref(stopper)) | holds_I(stop, veh_A, temp_ref(stop)) |

Rules (4) and (5) infer:

```
holds_I(combine(finir, course), author, temp_ref(finir))
  holds_I(combine(end, way), author, temp_ref(end))
holds_I(combine(percuter, author), veh_A, temp_ref(percuter))
  holds_I(combine(hit, author), veh_A, temp_ref(hit))
```

Finally, we obtain from rule (6) the literals:

```
holds_I(combine(combine(sur1, circuler1), voie1), author, temp_ref(sur1))
  holds_I(combine(combine(on1, drive1), lane1), author, temp_ref(on1))
holds_I(combine(combine(dans, dérapé), virage), author, temp_ref(dans))
  holds_I(combine(combine(in, skid), curve), author, temp_ref(in))
holds_I(combine(combine(sur2, dérapé), gravier), author, temp_ref(sur2))
  holds_I(combine(combine(on2, skid), gravel), author, temp_ref(on2))
```

¹⁴ Reports B2.A and B are written by the two protagonists of the same accident, who happened to be affiliated to the same insurance company.


```

holds_I(combine(combine(de2, tomber), author), author, temp_ref(de2))
holds_I(combine(combine(from, fall), author), author, temp_ref(from))
holds_I(combine(combine(sur3, finir), voie2), author, temp_ref(sur3))
holds_I(combine(combine(on3, end), lane2), author, temp_ref(on3))
holds_I(combine(combine(sur4, circuler2), voie3), veh_A, temp_ref(sur4))
holds_I(combine(combine(on4, drive2), lane3), veh_A, temp_ref(on4)).

```

5.4. Atemporal semantic literals

This second step deals with the semantic content, and aims at producing a version of semantic literals close to the final version: it differs from it only in the temporal parameters that remain unsolved at this stage. That is the reason why the literals resulting from this step are called *atemporal*.

This step effectively performs the convergence towards the limited set of concepts necessary to start the semantico-pragmatic reasoning. The inference rules active at this step (around 300 rules) capture, for each relevant concept, the different linguistic contexts which can lead to detect its presence. They associate also to each inferred concept the appropriate temporal reference which remains expressed by means of a word of the text.

In order to factor out what is common to several classes of words, and thereby to avoid being completely *ad hoc* at this stage, we have used lexical information to categorize words into semantic classes and types. This enables to group together rules that would otherwise be completely instantiated, i.e. the arguments of their premises would be merely words, and thus leads to a better level of abstraction.

Although most of the rules remain very dependent on our particular domain, there is however a small subset of rules that have a high level of generality and which can easily be transported to other domains (rules (7)–(10)). These rules capture the cases where there is a direct passage from a verb (transitive or intransitive) and its linguistic arguments at the linguistic level to a property and its parameters at the conceptual level. We have introduced a literal: `type (V, direct_trans)` for each verb *V* for which this characteristic holds. The binary predicate `sem_rep (P, Q)` (for *semantic representative*) holds when *Q* is the semantic constant (concept) associated to the word *P*.

- (7) `type (A, vehicle) ∧ type (P, direct_trans) ∧ intransitive (P) ∧ sem_rep (P, Q) ∧ holds_I (P, A, T) → holds (Q, A, T)`
- (8) `type (A, vehicle) ∧ type (P, direct_trans) ∧ intransitive (P) ∧ sem_rep (P, Q) ∧ ¬holds_I (P, A, T) → ¬holds (Q, A, T)`
- (9) `type (A, vehicle) ∧ type (B, object) ∧ type (P, direct_trans) ∧ transitive (P) ∧ sem_rep (P, Q) ∧ holds_I (combine(P, B), A, T) : holds (combine(Q, B), A, T) [¬qualif_v (P, passive)]`
- (10) `type (A, vehicle) ∧ type (B, object) ∧ type (P, direct_trans) ∧ transitive (P) ∧ sem_rep (P, Q) ∧ ¬holds_I (combine(P, B), A, T) : ¬holds (combine(Q, B), A, T) [¬qualif_v (P, passive)]`

Example (continued). The verbs *circuler* (to drive), *déraper* (to skid), *tomber* (to fall), *stopper* (to stop) and *percuter* (to hit) present in the text B2.B belong to the class of verbs that are directly translatable into semantic predicates. All these verbs are transitive. We have then the following literals:

| | |
|---|--|
| <code>type(circuler, direct_trans)</code> | <code>type(drive, direct_trans)</code> |
| <code>type(déraper, direct_trans)</code> | <code>type(skid, direct_trans)</code> |
| <code>type(tomber, direct_trans)</code> | <code>type(fall, direct_trans)</code> |
| <code>type(stopper, direct_trans)</code> | <code>type(stop, direct_trans)</code> |
| <code>type(percuter direct_trans)</code> | <code>type(hit, direct_trans)</code> |
| <code>transitive(percuter)</code> | <code>transitive(hit)</code> |
| <code>intransitive(circuler)</code> | <code>intransitive (drive)</code> |
| <code>intransitive(déraper)</code> | <code>intransitive (skid)</code> |
| <code>intransitive(tomber)</code> | <code>intransitive (fall)</code> |
| <code>intransitive(stopper)</code> | <code>intransitive (stop)</code> |

We have also the following predicates concerning the semantic classes of these verbs:

| | |
|--|---|
| <code>sem_rep(circuler, is_moving)</code> | <code>sem_rep(drive, is_moving)</code> |
| <code>sem_rep(déraper, is_slipping)</code> | <code>sem_rep(skid, is_slipping)</code> |
| <code>sem_rep(tomber, is_falling)</code> | <code>sem_rep(fall, is_falling)</code> |

```
sem_rep(stopper, Stop)          sem_rep(stop, Stop)
sem_rep(percuter, is_hitting)   sem_rep(hit, is_hitting)
```

Rules (7), (8) and (9) are applied here. Rule (7), which applies to intransitive verbs at the affirmative form, yields:

```
holds(is_moving, author, temp_ref(circuler1))
  holds(is_moving, author, temp_ref(drive1))
holds(is_slipping, author, temp_ref(déraper))
  holds(is_slipping, author, temp_ref(skid))
holds(is_falling, author, temp_ref(tomber))
  holds(is_falling, author, temp_ref(fall))
holds(is_moving, veh_A, temp_ref(circuler2))
  holds(is_moving, veh_A, temp_ref(drive2))
```

Rule (8) for transitive verbs at the negative form infers:

```
holds(Stop, veh_A, temp_ref(stopper))
  holds(Stop, veh_A, temp_ref(stop)).
```

The default (9), for transitive verbs at the affirmative form and the active voice, gives:

```
¬holds (combine(is_hitting, author), veh_A, temp_ref(percuter))
¬holds(combine(is_hitting, author), veh_A, temp_ref(hit))
```

The default (10) is not used in this example; it works for transitive verbs at the negative form and the active voice.

Similar rules are used for transitive verbs in the passive. In addition to these rules, which deal with the directly translatable verbs, we need more specific rules, see rules (11)–(12) in Appendix C.

5.5. Final semantic literals

The last thing to do, before obtaining the set of semantic literals that represent what the text evokes explicitly, is to replace temporal references still linked to words, by integers reflecting the linear order of the events described in the text. In general, the linear order of the events is not very far from the linear order of the narration. Our strategy is then to consider by default that these two orders coincide unless there are constraints that impose incompatibilities between them. So, we take the linear order of the narration as a starting point on which we bring the changes that are necessary to satisfy the temporal constraints obtained by appropriate inference rules.

5.5.1. Inferring temporal constraints

The temporal model adopted in our approach has been discussed in Section 4.5. Let us recall here our postulate: the linear order of the events described in the text is sufficient for our purpose. Moreover, our temporal model remains neutral as to whether the temporal parameter stands for an interval or for a time point. Consequently, the only relevant relations in our context are immediate precedence and simultaneity, expressed respectively by `prec`(R_1 , R_2) and `simul`(R_1 , R_2) where R_1 and R_2 are temporal references (we do not need all of Allen's relations [2]).

To extract automatically such temporal relations, we have written about 50 inference rules that exploit general linguistic markers, as well as common sense knowledge. Again, some inference rules are rather general and can be easily reused in other domains (e.g. those based on information about grammatical tenses, or about general common sense knowledge, such as “causes precede their effects”), while the others are much more dependent on the particular domain of car accidents (see rules (13)–(14) in Appendix C).

5.5.2. Resolving temporal constraints

At this stage, we have a set of atemporal semantic literals and a set of precedence and/or simultaneity constraints. We now sketch the algorithm that possibly alters the temporal structure obtained by merely following the narration, in order to satisfy the temporal constraints.

Let `ref`(M_1), ..., `ref`(M_n) be the temporal references involved in the atemporal literals built so far (the words M_1 , ..., M_n appear in the text in this order) and R_1 , ..., R_m the temporal constraints. Each reference `temp_ref`(M_i) is first associated to the integer i ; the goal is to find the integer T_i , which replaces at the end of the algorithm the provisional parameter `temp_ref`(M_i). We define a temporal structure S to be the sequence T_1 , ..., T_n . The general schema of the algorithm is as follows (more details are given in [53]):

Initial step: $T_i \leftarrow i$, for $1 \leq i \leq n$. (narrative order)

repeat

find the smallest k , $1 \leq k \leq m$ such that $\neg \text{satisfy}(S, R_k)$
 modify(S, R_k)

until all temporal constraints are satisfied.

Replace each $\text{temp_ref}(M_i)$ in an atemporal literal by the integer T_i .

where $\text{satisfy}(S, R_k)$ is trivially:

if R_k is $\text{prec}(\text{temp_ref}(M_j), \text{temp_ref}(M_i))$

then if $T_j = T_i - 1$ **return true**

else return false;

if R_k is $\text{simul}(\text{temp_ref}(M_j), \text{temp_ref}(M_i))$

then if $T_j = T_i$ **return true**

else return false;

and $\text{modify}(S, R_k)$ is:

if R_k is $\text{prec}(\text{temp_ref}(M_j), \text{temp_ref}(M_i))$

then $T_j \leftarrow T_i - 1$;

if R_k is $\text{simul}(\text{temp_ref}(M_j), \text{temp_ref}(M_i))$

then $T_j \leftarrow \max(T_j, T_i)$;

close the gap, i.e. renumber the T_i in such way that S is always a sequence on the (multi-) set of the first positive integers.

The choice of $\max(T_j, T_i)$ (the maximum of the two values T_i and T_j) has empirically proven to be a good heuristic.

Example (continued). The atemporal semantic literals inferred from the text B2.B were:

```
holds(is_moving, author, temp_ref(circuler1))
holds(is_moving, author, temp_ref(drive1))
holds(is_slipping, author, temp_ref(deraper))
holds(is_slipping, author, temp_ref(skid))
holds(cause_not_control, gravels, author, temp_ref(sur2))
holds(cause_not_control, gravels, author, temp_ref(on2))
holds(is_falling, author, temp_ref(tomber))
holds(is_falling, author, temp_ref(fall))
holds(bend, author, temp_ref(dans))
holds(bend, author, temp_ref(in))
holds(is_moving, veh_A, temp_ref(circuler2))
holds(is_moving, veh_A, temp_ref(drive2))
holds(Stop, veh_A, temp_ref(stopper))
holds(Stop, veh_A, temp_ref(stop))
holds(combine(is_hitting, author), veh_A, temp_ref(percuter))
holds(combine(is_hitting, author), veh_A, temp_ref(hit)).
```

The initial structure is thus S_1 (see Fig. 3).

Among the precedence relations inferred from our text, the relation $\text{prec}(\text{temp_ref}(\text{sur}_2), \text{temp_ref}(\text{deraper}))$ is not satisfied by S_1 . To satisfy this relation, we modify the temporal structure and get S_2 that satisfies all the precedence relations¹⁵ (see Fig. 4).

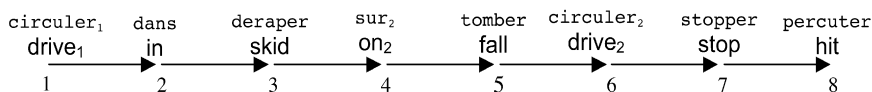
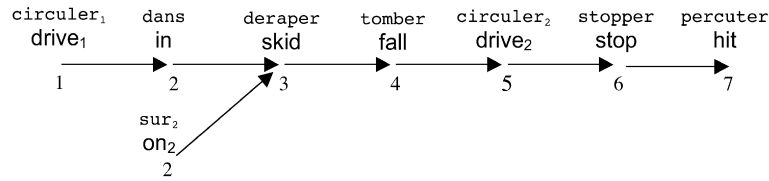


Fig. 3. Temporal structure S_1 .

¹⁵ In this example, one iteration of the algorithm is enough to satisfy the temporal constraints. However, for some texts of the corpus, several iterations are required to satisfy them all.

Fig. 4. Temporal structure S_2 .

The final semantic literals for text B2.B are then:

```
holds(is_moving, author, 1)
holds(bend, author, 2)
holds(cause_not_control, author, 2)
holds(is_slipping, author, 3)
holds(is_falling, author, 4)
holds(is_moving, veh_A, 5)
holds(Stop, veh_A, 6)
holds(combine(is_hitting, author), veh_A, 7)
```

5.6. Discussion

Our objective is to carry out a complete chain of treatment, going from the input text to the cause of the accident where NL understanding is achieved mainly by an inference process. The linguistic reasoning is the first step of this process. Since its role is to prepare the input of a second step, the semantico-pragmatic reasoning (see the next section), which represents the main contribution of this work, we developed it to the best we could without seeking generality. In order to generalize our approach to other domains, it would obviously be necessary to delve more deeply into lexical and syntactic phenomena, and to implement more generic solutions to the problems involved in the linguistic reasoning. This requires using general lexical, morphological, syntactic principles to design, at a meta-level, the abstract patterns of inference enabling to extract the relevant semantic information explicitly conveyed by a text.

6. The semantico-pragmatic reasoner

The semantic literals resulting from the previous stage are supposed to represent the relevant information explicitly evoked in the text. This information is considered to be “hard”, i.e. it is no longer defeasible. Unlike in formal semantics, where this kind of information is the main objective, in our case, semantic literals are just a starting point to the semantico-pragmatic reasoning that goes farther in simulating human understanding of NL. In fact, it has been often noticed that what is made explicit in a text is seldom sufficient to derive the conclusions that a human reader draws from that text. Human readers jump very easily to unwarranted conclusions, and they are generally not even conscious of doing it: most of the time, they are unable to tell what they have read from what they have derived [10]. But reproducing this capacity in a computer program is far from obvious!

To do so, we appeal to the norms of the domain, and we encode this knowledge again in non-monotonic inference rules. The semantico-pragmatic reasoning performs a kind of expansion process that adds more semantic literals. We distinguish three subsets of norm-based inference rules (see Fig. 5):

- a set of general rules, which add literals expressing properties that are likely to hold in the situation described,
- a set of rules for inferring duties that agents must respect to avoid a potential accident,
- and a set of rules to infer whether the agents had the capacity to respect their duties.

The expansion phase is followed by a process which aims to pick up from all the conclusions obtained so far, those which contribute to the determination of the cause of the accident. This latter process is a kind of convergence from the variety of predicates resulting from the analysis, towards a limited number of semantic literals (the *kernel*). In the following section, we discuss first some ontological elements used in the semantico-pragmatic reasoning, and then show the different steps of this process on our running example.

6.1. Some ontological elements

The semantico-pragmatic reasoning does not handle NL words; they have been replaced by “concepts” through the previous stages. There are different natures of concepts: in addition to the *state-dependent* predicates that represent the dynamic evolution of the situation, we need to represent some *state-independent* predications to define several static classes of the properties involved, and some static relations between them.

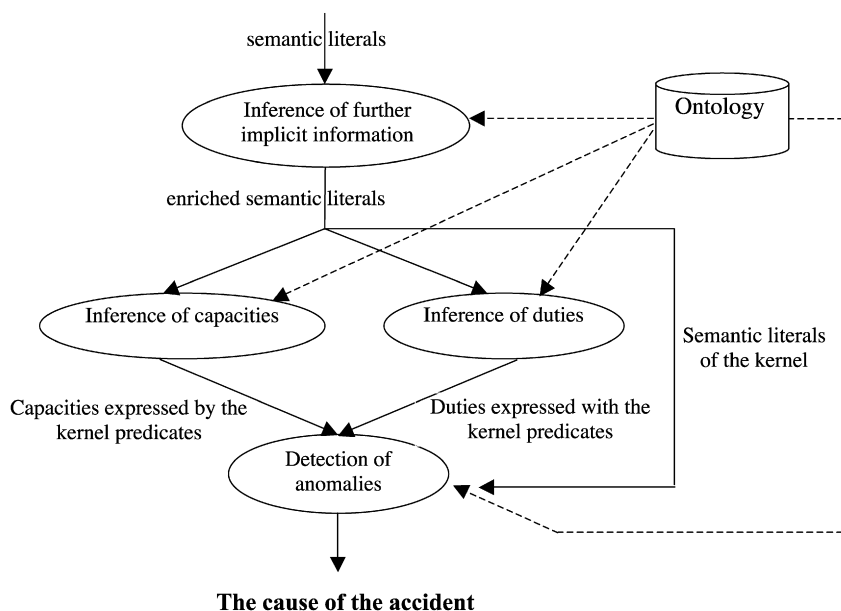


Fig. 5. Architecture of the semantico-pragmatic reasoning process.

Table 2

The literals of the kernel.

| | |
|--|---|
| holds (Stop, A, T) | A is stopped |
| holds (Drives_app, A, T) | A drives 'fairly' slowly, i.e. has a speed appropriate to the circumstances |
| holds (Control, A, T) | A has control over his/her vehicle |
| holds (Has_driver, A, T) | the vehicle of A has a driver |
| holds (Is_moving_off, A, T) | A moves off his/her vehicle |
| holds (Is_moving_backwards, A, T) | A moves backwards |
| holds (On_normal_lane, A, T) | A drives on his/her normal lane |
| holds (combine (Disruptive_factor, X), A, T) | X is a disruptive factor for A |

6.1.1. State-dependent predications

State-dependent predications are those whose truth-value varies with time. As we already discussed (Section 4.5), the scene of the accident is decomposed into several consecutive temporal states, characterized by the set of state-dependent predications holding at that state. Thus, in addition to some other predicates that will be described hereafter, the occurrences of the literals *holds* (P, A, T), *duty* (P, A, T) and *is_able_to* (P, A, T) represent the major part of the state-dependent predications. Let us now focus on a small but very important subset of these predications, which constitutes what we call the kernel.

The kernel is a set of 8 properties, in terms of which every cause must be expressed, and towards which the semantico-pragmatic reasoning process converges. The kernel literals are listed in Table 2.

The advantage of the kernel is twofold: it works, on the one hand, as a stopping criterion for the semantico-pragmatic reasoning, and on the other hand it ensures a kind of homogeneity in the concepts involved in expressing causes, which is helpful if we want to compare the answers of the system with those of human readers. The choice of the properties to put in the kernel is guided by the following principles:

Sufficiency: The properties of the kernel are sufficient to express the various propositions that can reasonably be considered as causes of the accidents in the whole corpus. There has been so far no need to introduce other properties to express any cause in the corpus.

Minimality: The number of properties in the kernel is minimal. No subset of the kernel is able to express all the causes evoked in the corpus.

Adequacy: The kernel properties are situated at a level of granularity, which corresponds fairly well to that of the answers expected from an ordinary human reader asked about the cause of an accident.

6.1.2. State-independent predications

The state-independent predications provide the variables occurring in the state-dependent predications with types. Types are used, together with other predicates, as premises of the semantico-pragmatic inference rules for two reasons: it increases the level of abstraction, hence it improves the generality of the rule, and it limits the set of concepts on which rules are applied, by limiting the range of each variable. The main classes considered in our system are now discussed.

Effects

We mean by effects the set of properties that characterize the state of an agent at a given time. Indeed, an agent's state changes as soon as the truth-value of at least one effect changes. Stopping, driving, controlling one's vehicle, or bumping into an obstacle are some examples of effects. The literal `effect (F)` expresses that `F` is the name of an effect.

Some effects are persistent by default, i.e. their truth-value is maintained forward in time, as long as no action and no event alters it. This is the usual way to handle the famous “*frame problem*”, known since McCarthy and Hayes [46] and very abundantly described in the literature ever since. The predicate `persistent_effect(F)` denotes that `F` is a persistent effect, and a simple default rule states that, as long as this is consistent, if `F` is a persistent effect and `holds (F, A, T)` is true (resp. is false), then so is `holds (F, A, T+1)`.

Actions

The literal `action (Act)` is true if and only if `Act` is the name of an action that an agent may perform voluntarily in order to achieve an effect. Braking, turning the steering wheel are examples of actions that normally lead respectively to the effects of stopping and turning right or left. In our context, the number of actions is very limited. An agent may sometimes want to maintain his/her current state, especially when an external event occurs which leads to an undesirable state. For that purpose, a special action is at the disposal of the agent: for every effect `F`, `combine (Keep_state, F)` is an action.

(15) `effect (F) → action (combine (Keep_state, F))`

Events

The literal `event (Evt)` is true if and only if `Evt` is the name of an event. From the agent's standpoint, whatever occurs independently of his/her will is considered as an event. For instance, the outbreak, in the proximity of an agent, of an obstacle (e.g. a tree, an animal, or a vehicle) or of some object that can lead him/her to lose the control of his/her vehicle (for example, gravels, icy patch, or oil) are considered as events.

Kernel

The fact for a predicate `P` to belong to the kernel (see above) is expressed in the language by the literal `kernel (P)`.

Relations between concepts

We list in a database a number of time-independent relations between concepts. Let us examine the main relations:

To capture the fact that two effects `F` and `F'` can never be fulfilled simultaneously, i.e. in the same temporal state, we use the literal `incompatible (F, F')`. A special case of this incompatibility relation concerns the case of an effect and its negation. Thus we have:

(16) `effect (F) → incompatible (F, not(F))`

We list also in a database a number of facts of the form `pot_cause (P, F)` where `F` is an effect and `P` is either an action or an event. As explained above (Section 4.6), this relation means that `P` is the cause that comes most probably to the mind of an ordinary reader, when s/he learns that `F` occurred. Moreover, the limitation of the domain allows us to add the following simplifying hypothesis, for which we have found as yet no exception, but the one given hereafter:

Hypothesis. For every effect `F`, at most one action `Act` satisfies `pot_cause(Act, F)`.

Exception. If `F` is persistent, the above-mentioned action of keeping state `F` is also a potential cause of `F`:

(17) `persistent_effect (F) → pot_cause (combine(Keep_state, F), F)`

For an action `Act` to reach an effect `F`, which it is known to be a potential cause of, the agent must perform it under appropriate circumstances (this is the *qualification* problem already mentioned). As we neither want to, nor can, list these circumstances, we use the default assumption that actions are performed in situations where they are expected to succeed. We write `precond_act (F, P)` to express that `P` must be true for the (supposedly unique) action able to yield effect `F` to succeed. The assumption is thus expressed by a default, which is also a case where state-independent predications are used to derive state-dependent ones:

(18) `holds (F, A, T) ∧ precond_act (F, P) : holds (P, A, T-1)`

Symmetrically, we write: `precond_avoid_event (F, P)` for: `P` must be true for an agent to succeed in *avoiding* the effect `F` of an event.

6.2. Inference of further implicit information

Enriching the explicit content of a text by implicit information is by no means a new idea: *frames* [49] and *scripts* [58] have been proposed in the same spirit, but at the time they have been developed, the tools to handle them properly had not been invented, and they suffered from a lack of formalization. The remarkable advances of AI in terms of modeling non-monotonic reasoning now enables a more adequate account of the knowledge about norms to be given. We have explained above (Section 4.4) why semi-normal defaults are good candidates to express the norms we need to cope with.¹⁶

The inferred propositions have different natures. They may concern the agents' duties and capacities (see the next two sections) but they may also express the fulfillment of some properties of agents in different temporal states of the scene. We have developed for this last kind of reasoning around 200 inference rules, concerning various topics of the road domain, e.g.:

- lines of traffic,
- control of vehicles and its loss,
- predictability of obstacles,
- priority between vehicles at intersections, ...

The reader will find in Appendix C (rules (19)–(29)) how the running example is treated by these rules.

6.3. Inference of duties

In order to determine the cause of an accident, we must know who had the duty to do some actions, and when. For instance, we must have the knowledge that one must stop in presence of a red light, one should not move off while there are other vehicles in front ... The respect of a duty by an agent represents obviously an important norm:

(N_1) “In general, we expect that agents respect their duties”

According to our conception of causality, the determination of anomalies is based on the hypothesis that if an accident occurs, then there must be either some external abnormal disturbing factor that can explain the accident, or an agent who did not respect a norm. In the latter case, anomalies inevitably involve a violation of an occurrence of the abstract norm (N_1). Other norms have various strengths, depending on which they can be taken as primary causes of accidents or not. For example, in a situation where after a traffic light turns green, a first vehicle does not move off, and a second one bumps into it, two norms are violated, (N_2) and (N_3):

(N_2) “A vehicle should not move off as long as its preceding vehicle did not move off”

(N_3) “A vehicle stopped at a red light should move off when the light turns green”

However, we prefer as a cause of the accident the violation by the second vehicle of the norm (N_2). Actually, (N_3) describes an expected behavior, and its violation is no excuse for the driver of the second vehicle. The obstacle created by the failure of the first one to respect (N_3) does not dispense him/her from respecting (N_2), and failing to do so is perceived as the ‘real cause’. So our system should find the violation of (N_2) as the primary anomaly.

Expressing the agent's duties has required around 25 inference rules. Those that apply to the running example are rules (30)–(33) in Appendix C.

6.4. Inference of capacities

A crucial point for detecting whether an anomaly (the violation of a norm) is primary or derived consists in assessing whether an agent is in position to avoid a transition yielding an undesired state, i.e. if s/he *is_able_to* produce or prevent some given effect. More information about the features of the actions and events is needed to define properly this predicate. We have developed around 25 inference rules to deal with the capacities of agents. Unlike the rules described in the two previous sections, relatively dependent on the road domain, the rules concerning capacity are characterized by a higher degree of abstraction and could easily be reused in other domains.

The literal *predictable* (V , A , T) expresses a property of an event V that can be state-independent (e.g. *icy patches are considered as unpredictable causes of loss of control*); it can also be inferred in specific situations by means of appropriate rules (e.g. *if X is an obstacle for A , and X is not under control, then the behavior of X is considered unpredictable for A*).

An event V is said to be *controllable* by agent A at time T iff either V does not occur at time T , or it was predictable, and A is in position at time T to satisfy the precondition of its avoidance.

¹⁶ To preserve the efficiency of the computations, we keep the number of default rules as small as possible. This means that even if the knowledge represented is known to have exceptions, as long as in the restricted domain of road accidents these exceptions are unlikely to occur, we express it with a material implication.

```

(34) Event (V) ∧
    ( ¬holds (V, A, T)
      ∨ ( predictable (V, A, T)
          ∧ (precond_avoid_event (F, P) → holds (P, A, T))
        )
    ) ↔ controllable (V, A, T)

```

An agent may undertake an action *Act* without knowing whether it will succeed. The predicate *available* is meant to express that, as far as the agent knows, *Act* satisfies all its preconditions. The literal *available* (*Act*, *F*, *A*, *T*) is thus true iff if at time *T*, agent *A* decides to execute *Act* with the belief that effect *F* will obtain. A default assumption is that every action is available:

```

(35) pot_cause (Act, F) ∧ action (Act) : available (Act, F, A, T)

```

This assumption forces to enumerate the situations where an action is not available. Several cases of unavailability are considered:

- the presence of “technical problems”,
- the precondition of the action not being satisfied,
- a keep-state action is available except if an uncontrollable event leads to a state where an effect *F'* holds, and *F'* is incompatible with *F*,
- the loss of control of a vehicle makes every action of its driver obviously unavailable.

These cases are respectively expressed by the rules:

```

(36) holds (combine (Tech_Pb, Act), A, T) ∧ holds (Act, A, T) ∧ pot_cause (Act, F) →
    ¬available (Act, F, A, T)

```

```

(37) action (Act) ∧ precond_act (F, P) ∧ ¬holds (P, A, T) → ¬available (Act, F, A, T)

```

```

(38) (∃ F', V) (pot_cause (V, F') ∧ event (V) ∧ ¬controllable (V, A, T) ∧
    Incompatible (F, F')) ↔ ¬available (combine (Keep_state, F), F, A, T)

```

```

(39) ¬holds (Control, A, T) ∧ pot_cause (Act, F) → ¬available (Act, F, A, T)

```

These predicates delimit the states where an agent *is_able_to* undertake an action. Intuitively, agent *A* is able to reach effect *F* at time *T* iff there exists an action *Act* that is a potential cause for *F* and is available for *A* at *T*:

```

(40) is_able_to (F, A, T) ↔ (∃ Act) (action (Act) ∧ available (Act, F, A, T) ∧
    pot_cause (Act, F))

```

Example (continued). At the previous stages, some duties have been derived at temporal states *T*, as well as the fact that the corresponding effects did not obtain at states *T* + 1. These pairs are:

```

duty (Control, author, 2), ¬holds (Control, author, 3)
duty (Has_driver, author, 4), ¬holds (Has_driver, author, 5)
duty (Stop, author, 6), ¬holds (Stop, author, 7)

```

These pairs are the sign of anomalies. In order to know whether they are primary (i.e. possible causes of the accident) or not, what remains to be seen is whether the agent was able to comply with his/her duties.

Let us start by looking for the information about the capacity of the agent *author* to keep control at temporal state 2: this depends on the availability of the action *combine*(*Keep_state*, *Control*) for him/her at that state. Since the loss of control resulted from the occurrence of an event (the presence of gravels on the road), the availability of this action depends on whether this event was or not controllable by the agent. Rule (34) provides this information; we already have:

```

predictable (combine(cause_not_control, gravels), author, 2)
¬holds (Drives_app, author, 2)

```

Moreover, we have the static relation (41), which expresses that driving slowly is a precondition to be able to avoid the effect of an event leading to lose control:

```

(41) object(X) → precond_avoid_event (combine(cause_not_control, X), not(Control),
    Drives_app)

```


with $X = \text{gravels}$, we get: $\text{precond_avoid_event}(\text{combine}(\text{cause_not_control}, \text{gravels}), \text{not}(\text{Control}), \text{Drives_app})$.

In this situation, the precondition is not fulfilled. Therefore rule (34) infers: $\neg\text{controllable}(\text{combine}(\text{cause_not_control}, \text{gravels}), \text{author}, 2)$.

Applying rule (38) with $V = \text{combine}(\text{cause_not_control}, \text{gravels})$, $F = \text{Control}$, $F' = \text{not}(\text{Control})$, $A = \text{author}$, $T = 2$, we obtain: $\neg\text{available}(\text{combine}(\text{Keep_state}, \text{Control}), \text{Control}, \text{author}, 2)$.

With this conclusion, we deduce by (40): $\neg\text{is_able_to}(\text{Control}, \text{author}, 2)$.

For the second pair, the capacity at time 4 of the agent *author* to be inside his/her vehicle depends on the information about the controllability of the ejection of the author. We have already inferred that this event is not predictable at time 4: $\neg\text{predictable}(\text{is_falling}, \text{author}, 4)$.

Consequently, we can deduce by rule (34) that: $\neg\text{controllable}(\text{is_falling}, \text{author}, 4)$.

Analogously to the previous case, we obtain the literal:

$\neg\text{available}(\text{combine}(\text{Keep_state}, \text{Has_driver}), \text{Has_driver}, \text{author}, 4)$

And consequently the result: $\neg\text{is_able_to}(\text{Has_driver}, \text{author}, 4)$.

The last pair requires assessing the capacity of the agent *author* to stop at time 6. To do so, we have to seek whether the action of braking is available for the author at time 6 or not.

The fact that the agent *author* has no control at state 6 triggers the following rule (42):

(42) $\text{holds}(\text{Control}, A, T) \wedge \text{pot_cause}(\text{Act}, F) \wedge \text{kernel}(F) \rightarrow \neg\text{available}(\text{Act}, F, A, T)$

This rule allows to infer ($A = \text{author}$, $T = 6$, $\text{Act} = \text{brake}$, $F = \text{Stop}$): $\neg\text{available}(\text{brake}, \text{Stop}, \text{author}, 6)$.

Eventually, rule (40) provides us with what we wanted to know: the agent *author* was not able at state 6 to stop in order to avoid the collision: $\neg\text{is_able_to}(\text{Stop}, \text{author}, 6)$.

6.5. Detecting anomalies

Having enriched the initial set of semantic literals by implicit information, the last stage of the semantico-pragmatic reasoning process consists in selecting the elements necessary to detect the primary as well as the derived anomalies. It may be the case, however, that these elements do not lead to a primary anomaly but only to derived ones. In such cases, the primary anomaly is often situated in former temporal states and corresponds to a precondition which does not hold, while being necessary for the success of an action leading to avoid the accident. Some other inferences are then applied in order to solve this case.

For the detection of anomalies, we have written 8 inference rules. We give hereafter the formal description of primary and derived anomalies introduced informally in Section 4.3 and we show on our running example the sequence of inferences that enable our system to detect a primary anomaly.

As said earlier, there are two forms of primary anomalies. The first one (43) corresponds to the case of an agent *A* having the duty and the capacity at time *T* to achieve some effect *F*, but for which we observe that another effect *F'* incompatible with *F* holds at time *T*+1.

(43) $\text{duty}(F, A, T) \wedge \text{is_able_to}(F, A, T) \wedge \text{holds}(F', A, T+1) \wedge \text{incompatible}(F, F') \rightarrow \text{Primary_anomaly}(F, A, T+1)$

The second form of primary anomaly (44) is detected when some disturbing event happens independently of the will of agents, and this event can alone explain the accident. As an example of such events, called disruptive factors, we mention the presence on the road of an unpredictable oil puddle leading to a loss of control, and consequently to an accident. More generally, all unpredictable causes of loss of control are considered as disruptive factors¹⁷ (when they effectively lead to accidents), and so are the unpredictable obstacles other than vehicles (e.g. dogs) as well as the technical problems arising while executing actions necessary to avoid accidents.

(44) $\text{holds}(\text{combine}(\text{Disruptive_factor}, C), A, T) \rightarrow \text{Primary_anomaly}(C, A, T)$

Derived anomalies (45) correspond to situations where the reason of an agent for not respecting his/her duty is that s/he is not *able* to do so. Thus, the only difference between derived anomalies and the first form of primary anomalies lies in the polarity of the modality *is_able_to*:

(45) $\text{duty}(F, A, T) \wedge \neg\text{is_able_to}(F, A, T) \wedge \text{holds}(F', A, T+1) \wedge \text{incompatible}(F, F') \rightarrow \text{Derived_anomaly}(F, A, T+1)$

¹⁷ We list in a static database the different causes of loss of control with the information about their predictability.

Appendix C develops the last steps enabling the system to derive:

```
Derived_anomaly (Control, author, 3)
Derived_anomaly (Has_driver, author, 5)
Derived_anomaly (Stop, author, 7)
Primary_anomaly (Drives_app, author, 1)
```

The conclusion can be expressed as follows:

"The cause of the accident is that, at the moment when the author entered the road bend, s/he did not drive at an appropriate speed".

6.6. Discussion

We have shown in details how to go from the text, through the linguistic literals, the semantic literals, the kernel, eventually to an action that some agent had the duty to do, was able to do, and did not do. The reader has certainly noticed that the rules used have various degrees of generality, some of them looking really *ad hoc* for the text under consideration. As the validation section (Section 8 below) will show, this impression does not correspond to reality: most of the texts, very different from those we had at the time where the rules were written, are nonetheless correctly processed by these rules. The fact remains however that writing rules at a more abstract, homogeneous, level would be by far preferable. But we feel comforted by the history of the field in trying not too soon to discover general rules: fulfilling local goals seems a safer way to find fruitful paths, rather than looking straightway for universal reasoning patterns.

7. Implementation issues

This section discusses implementation issues. We give first a brief overview of the answer set programming paradigm; then we explain how to transform the inference rules presented above into extended logic programs [54].

7.1. Answer Set Programming

Answer Set Programming (ASP) is a recent paradigm covering different kinds of logical programming, and their semantics [44]. It allows representing and solving various standard problems in Computer Science: combinatorial problems as k-coloring graph, path finding, scheduling. But ASP is also concerned by problems arising in Artificial Intelligence, when the available information is incomplete, as in non-monotonic reasoning, planning, diagnosis, ... The reader may find additional information about ASP on the web site of the working group WASP (<http://wasp.unime.it/>).

Here, we are particularly interested in using ASP as a framework for default reasoning. For this purpose, we use Extended Logic Programs (ELP) to represent knowledge by means of rules containing positive information and strong or default negative information, and we interpret them by answer set semantics [22]. Formally, an ELP is a set of rules of the form:

$$c \leftarrow a_1, \dots, a_n, \text{ not } b_1, \dots, \text{ not } b_m \quad (n \geq 0 \text{ and } m \geq 0)$$

where c , a_i and b_j are literals.¹⁸

For a given rule $r = c \leftarrow a_1, \dots, a_n, \text{ not } b_1, \dots, \text{ not } b_m$, we denote:

$$\text{head}(r) = c \quad \text{body}^+(r) = \{a_1, \dots, a_n\} \quad \text{body}^-(r) = \{b_1, \dots, b_m\} \quad r^+ = c \leftarrow a_1, \dots, a_n$$

Definition. Let R be a set of rules without default negation (i.e. for all $r \in R$, $\text{body}^-(r) = \emptyset$), R is called a *Definite Logic Program*.

A set of literals X is *closed* w.r.t. R when $r \in R$, $\text{body}^+(r) \subseteq X \Rightarrow \text{head}(r) \in X$.

The set of *consequences* of R is $C_n(R)$, the minimal set of literals that is consistent and closed w.r.t. R , or if such a set does not exist, it is equal to the whole set of literals of the language.

For a given set of literals A and an ELP P , the *reduct* of P by A is the definite Logic Program:

$$P^A = \{r^+ \mid r \in P \text{ and } \text{body}^-(r) \cap A = \emptyset\}$$

Let P be an ELP and A a set of literals. A is an *answer set* of P if and only if $A = C_n(P^A)$.

Examples.

$P1 = \{a \leftarrow \text{not } b, b \leftarrow \text{not } a, \neg c \leftarrow b\}$ has two answer sets $\{a\}$ and $\{b, c\}$

$P2 = \{a \leftarrow \text{not } a\}$ has no answer set at all

¹⁸ Since c , a_i and b_j are literals, they can be positive or negative, i.e. of the form p or $\neg p$ where p is an atom. There is an important distinction between the "hard" negation of a proposition ($\neg p$) and its negation as failure (not p). Hence, the expression not $\neg p$ is not equivalent to p .

Table 3

Translation of classical formulas into ELP.

| Classical formulas | ELP |
|---|--|
| a fact: a | a rule with empty body: a . |
| a conjunction of facts: $a_1 \wedge \dots \wedge a_n$ | n rules with empty bodies: $a_1 \dots a_n$. |
| a material implication: $a_1 \wedge \dots \wedge a_n \rightarrow b$ | one direct rule: $b \leftarrow a_1, \dots, a_n$ plus n contrapositives: $\neg a_1 \leftarrow \neg b, a_2, \dots, a_n$ \dots $\neg a_n \leftarrow \neg b, a_1, \dots, a_{n-1}$ |

Table 4

Translation of default rules into ELP.

| Classical formulas | ELP |
|--|---|
| a normal default: $A_1, \dots, A_n : B$ | a rule: $B \leftarrow A_1, \dots, A_n, \text{not } \neg B$. |
| a semi-normal default: $A_1, \dots, A_n : B [C_1, \dots, C_m]$ | a rule: $B \leftarrow A_1, \dots, A_n, \text{not } \neg B, \text{not } \neg C_1, \dots, \text{not } \neg C_m$. |

We have recalled the basic notions of answer set semantics only in the case of propositional rules. But obviously, a flexible knowledge representation requires the rules to contain variables. In this case, a rule is considered as a global schema for the set of its full instances, obtained by replacing every variable by every constant in the language.¹⁹

It is important for our work to point out that answer set semantics for *ELP* can be viewed as a subcase of default logic [22,6] by the translation of every rule $r = c \leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_m$ into the default rule:

$$T(r) = a_1 \wedge \dots \wedge a_n : c[\neg b_1, \dots, \neg b_m]$$

As a matter of fact, if S is an answer set of an *ELP* P , then $Th(S)$ is an extension of the default theory $(\emptyset, T(P))$. Conversely, every extension of $(\emptyset, T(P))$ is the deductive closure of an answer set of P .

Obviously, in whole generality, every default theory cannot be translated into an *ELP*. But as will be shown below, the restricted default theories corresponding to our representation language are translatable into *ELPs*. We can thus take advantage of the software packages for *ASP* that are available today.²⁰

7.2. From default rules to extended logic programs

In this section, we explain how to encode our knowledge base, originally expressed as a default theory, into an extended logic program. A very important point to note is that our original knowledge base does not contain disjunctions. Since a default theory is a pair consisting in a set of classical formulas and a set of default rules, we distinguish two types of translation represented respectively in Tables 3 and 4.

We have encoded our rules in default logic instead of using directly *ASP*, because default logic is more compact than *ASP*, which needs more rules, especially for contrapositives. The translation of default logic into *ASP* is easy to perform automatically.

7.3. The implementation

The program we have written contains a declarative part as well as an imperative one. The declarative part contains the set of the non-monotonic inference rules (about 615 inference rules altogether) for the linguistic and the semantico-pragmatic reasoning. The imperative part (around 8000 lines of code) represents the implementation in C of the parser, the post-treatment heuristics, and the resolution of the temporal constraints.

To give a rough idea of the behavior of the system in terms of the execution time,

- the linguistic reasoning requires an execution time varying in a broad fork, depending on the size of the text to be analyzed: from 1 second to 4 minutes and 56 seconds on CPU Pentium (3.0 GHz) and 1 Go of RAM.
- the situation is very different for the semantico-pragmatic reasoning: the execution time is in the interval between 3 and 8 seconds.

¹⁹ Our implementation is based on SMOELS [59], an answer set programming language. As SMOELS works for propositional rules, our inference rules are first treated by LPARSE, a tool that, among other things, instantiates the variables of the inference rules by values in their respective domains of definition to produce propositional rules. SMOELS and LPARSE are available at the web page: <http://www.tcs.hut.fi/Software/smodels>.

²⁰ In addition to SMOELS, other tools for answer set programming have been developed. For example: DLV [41] <http://www.dbai.tuwien.ac.at/proj/dlv>, and Cmodels [43] <http://www.cs.utexas.edu/users/tag/cmodels.html> *Nomore++* [3] <http://www.cs.uni-potsdam.de/wv/nom>.

8. Validation

8.1. Methodology

We consider two levels of validation: the first one consists in evaluating to which point the system we built is consonant with our own intuition concerning causality. As we said earlier, the corpus of 160 texts is made of reports for which we considered the cause of the accident to be relatively uncontroversial, insofar as the report is honest. The corpus has two parts: the training sample (73 texts) used to develop our inference rules, and the validation sample (87 texts) kept to assess how our system performs when applied to new texts. We report the results in the next section.

The second level, in a way the most valuable, consists in checking whether the output of the system fits with the answers provided by human readers other than ourselves, when they are asked: “what is the cause of the accident?”. As their answers are dispersed, the objective of the validation is more difficult to spell out: should the system give the most frequent answer (even if this frequency is below 50%)? Or a kind of centroid of all answers (if a distance can be defined)? Or to reproduce the probability distribution of the answers? In any case, we report the results of the experiment in Section 8.3.

8.2. Validation w.r.t. our intuition

The validation considered in this subsection consists in testing the conformity of the answers given by the system with our own answers.

As discussed earlier, the different parts of the system have not the same importance: the morpho-lexical and the syntactic analyzers have been developed in an ad hoc way, mainly to ensure a complete chain of treatment; therefore they will not be considered in the validation process. The main part of our work is the semantico-pragmatic reasoning, and this is where the results on the validation corpus matter. Another interesting test consists in assessing the degree to which the semantico-pragmatic reasoner tolerates the errors made by the previous steps of the system.

The most significant measure concerns the overall behavior of the system, i.e. how frequently the answer given by the system when it receives as input a text of the corpus coincides with our own judgment. But we are also interested in the evaluation of intermediary results. Fig. 6 represents the hierarchy of the tests performed: test 1 corresponds to the evaluation of the post-treatment heuristics (Section 5.1), tests 2.1 and 2.2 correspond to the evaluation of the linguistic reasoning (Sections 5.3 to 5.5) respectively without and with manual corrections of the results output by the previous steps. Tests 3.1 and 3.2 are analogous to tests 2.1 and 2.2 and concern the semantico-pragmatic reasoning (Sections 6.2 to 6.5). Test 3.1 is the most important one, because it corresponds to the global behavior of the system.

8.2.1. Test of the post-treatment heuristics

There are 346 subject/verb relations generated by the parser in the training sample, and 505 of them in the validation sample. Table 5 gives the number and the percentage of subject/verb errors detected and those properly corrected with

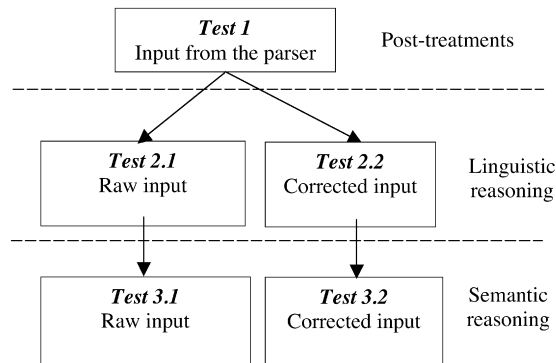


Fig. 6. Methodology of the test.

Table 5

Detection and correction of subject/verb errors.

| | Training sample | Validation sample |
|--------------------------------------|-----------------|-------------------|
| (1) Number of subject/verb relations | 346 | 505 |
| (2) Errors made by the parser | 34 (10%) | 37 (7%) |
| (3) Errors detected by the heuristic | 28 | 25 |
| (4) Errors correctly repaired | 26 | 20 |
| Recall (4)/(2) | 76% | 54% |
| Precision (4)/(3) | 92% | 80% |

Table 6

Success rate in anaphora resolution.

| | Training sample | Validation sample |
|---------------------------------|-----------------|-------------------|
| (1) Number of anaphors | 428 | 592 |
| (2) Number of referents found | 428 | 592 |
| (3) Number of correct referents | 424 | 564 |
| Recall (3)/(1) | 99% | 95% |
| Precision (3)/(2) | 99% | 95% |

Table 7

Success rate of the inference of the semantic literals.

| | Training sample | | Validation sample | |
|--|-----------------|-----------------|-------------------|-----------------|
| | Raw input | Corrected input | Raw input | Corrected input |
| (1) Literals expected | | 306 | | 326 |
| (2) Literals found | 296 | 298 | 278 | 279 |
| (3) Literals found with an incorrect predicate name | 0 | 0 | 4 | 4 |
| (4) Literals missing = (1) – [(2) – (3)] | 10 | 8 | 52 | 51 |
| (5) Literals found with the correct predicate name but an incorrect temporal parameter | 5 | 5 | 14 | 14 |
| (6) Literals found with the correct predicate name but an incorrect agent parameter | 3 | 0 | 11 | 0 |
| (7) Literals found and correct = (2) – [(3) + (5) + (6)] | 288 | 293 | 249 | 261 |
| Recall (7)/(1) | 94% | 96% | 76% | 80% |
| Precision (7)/(2) | 97% | 98% | 90% | 94% |

respect to the total number of errors (some errors are detected but wrongly corrected by the system) in the training and the validation samples.

As the table shows, the heuristics succeed in correcting three quarters of the erroneous relations in the training texts and more than one half of them in the validation texts. However, we will see later that the remaining errors do not lead to a worsening in the global performance of the system. We notice that the heuristics never over-corrects: the matrix never mistakenly detects an incompatibility between the subject and the verb of a correct relation. Let us now examine the results of the anaphora resolution heuristic (Table 6).

By construction, the heuristic finds a referent for every anaphor, hence the recall and the precision have the same value. For both the training and the validation samples, these values are very satisfactory. It is clear that this good performance is mostly due to the specificity of the domain and to the limitation of the kind of references that need be considered.

Looking at a higher level, we note that 20% of the texts in the validation corpus still contain at least one subject/verb error after applying the post-treatment heuristics, and 22% of them contain at least one incorrect reference. 31 texts (36%) contain either subject/verb or anaphora errors.

8.2.2. Test of the linguistic reasoner

To evaluate the linguistic reasoner, we have determined manually for each text the set of semantic literals we expect to be extracted from it. We then compare this set with the semantic literals inferred by the linguistic reasoner at two levels of granularity: literals and texts. What ensures to some extent that the manual determination of the “correct” set of the semantic literals as a base for the comparison does not affect the objectivity of the validation, is that the set of semantic predicates had already been determined at this stage of the development, and has not been built *ad hoc*. The total number of the expected semantic literals is 306 in the training sample and 326 in the validation sample.

Level of literals

A literal contains a predicate name and arguments. We have measured separately the number of literals having the correct predicate name (but possibly with wrong arguments) and the fully correct literals (name and arguments). The column “corrected input” corresponds to texts for which we have manually rectified the mistakes made by the previous stages.

Table 7 gives the results obtained by the linguistic reasoner in inferring the semantic literals. For the training texts, all the predicate names found are correct. For the validation texts, only 4 of them are erroneous. Giving a correct input rather than the raw result of the previous stages makes no significant difference. This means that the errors of the post-treatment heuristics have not a deep influence on the linguistic reasoner.

The table provides further details about the different kinds of errors made by the linguistic reasoner. The error rate remains rather limited in general and its peak concerns the inability for the linguistic reasoner to infer some semantic literals (in the validation sample 52 literals, out of the 326 that we expected, i.e. 16% of them have not been inferred). Although these errors do not necessarily affect the following steps of the system, this remark incites us to improve in our future work the generality of the linguistic rules by working at a more abstract level. Once again, there is no significant difference resulting from correcting or not the input of the linguistic reasoner. This confirms the fact that, fortunately,

Table 8

Synthesis of results of the linguistic reasoning at the text level.

| | Training sample | | Validation sample | |
|--|-------------------|-------------------|-------------------|-------------------|
| | Raw input | Corrected input | Raw input | Corrected input |
| Number of texts | | 73 | | 87 |
| Texts without errors | 60 (82%) | 63 (86%) | 35 (40%) | 39 (45%) |
| Texts with expected literals not found | 8 (11%) | 5 (7%) | 38 (44%) | 37 (40%) |
| Texts with additional literals | 0 | 0 | 4 (5%) | 4 (5%) |
| Texts with literals having erroneous non-temporal parameters | 2 (3%) | 0 | 8 (9%) | 0 |
| Texts with literals having erroneous temporal parameter | 5 (7%) | 5 (7%) | 12 (14%) | 12 (14%) |

Table 9

Error rate of the semantico-pragmatic reasoning.

| | Training sample | | Validation sample | |
|--|-----------------|-----------------|-------------------|-----------------|
| | Raw input | Corrected input | Raw input | Corrected input |
| (1) Expected primary anomalies | | 75 | | 88 |
| (2) Number of primary anomalies found | 76 | 75 | 89 | 90 |
| (3) Correct primary anomalies found | 71 | 74 | 75 | 83 |
| Recall for primary anomalies (3)/(1) | 95% | 99% | 85% | 94% |
| Precision for primary anomalies (3)/(2) | 93% | 99% | 84% | 92% |
| (6) Expected derived anomalies | | 48 | | 24 |
| (7) Number of derived anomalies found | 47 | 48 | 17 | 22 |
| (8) Correct derived anomalies found | 46 | 48 | 16 | 21 |
| Recall for derived anomalies (8)/(6) | 96% | 100% | 67% | 87% |
| Precision for derived anomalies (8)/(7) | 98% | 100% | 94% | 95% |

the subject/verb errors that remained after the application of the post-treatment heuristics are not very relevant for the inference of the expected semantic literals.

Text level

Table 8 displays the same results at the text level. We consider that a given text is correctly analyzed if and only if it contains no error or omission in any of its expected semantic literals. Table 8 shows that the linguistic reasoner errors are not concentrated in a part of the corpus: more than one half of the texts contain at least an error. This remark is crucial for the evaluation of the semantico-pragmatic reasoner (see Section 8.2.3 below), which must be able to find correct answers even in the presence of some errors in its input.

Let us now see the effect of the subject/verb errors and of unresolved anaphors on the performance of the linguistic reasoner at the text level. Remember that 31 texts of the validation corpus contain either one or the other of these mistakes:

- Without correcting the input of the linguistic reasoner, 10 of these 31 texts (30%) have no error in their inferred semantic literals.
- The correction of the input of the linguistic reasoner avoids the errors in 10 of the 21 remaining texts. In the 11 other texts, the semantic literals do not change. Thus, in 21 of these 31 texts (70%), the errors in the subject/object relations and in anaphora references have no effect on the linguistic reasoning.

8.2.3. Test of the semantico-pragmatic reasoner

This section discusses the results of the semantico-pragmatic reasoner, i.e. the determination of the primary and derived anomalies, which reflect respectively the most plausible cause of the accident and the other possible causes resulting from the first cause. The performance of the system is evaluated according to two factors: its capacity to treat new texts correctly, and its robustness, i.e., its capacity not to be influenced by the errors accumulated by the previous steps. The percentages given in the following tables are based on a manual analysis according to which 75 (resp. 88) primary anomalies and 48 (resp. 24) derived anomalies should have been detected in the training (resp. in the validation) sample (Table 9).

Anomaly level

For the *training texts*, the system determines directly (without correcting the input received from the linguistic reasoner) 95% of the expected primary anomalies. When the input is correct, the system finds 99% of the expected primary anomalies.

More significant is the percentage of the primary anomalies directly found in the *validation texts*: 85%. This value is very satisfactory and shows that our system, as a whole, is able to cover a wide range of phenomena that are relevant to the detection of the causes in car crash reports. Moreover, correcting the semantic literals obtained from the linguistic reasoner enables the system to obtain as much as 94% of the primary anomalies expected in the validation corpus. This very good percentage reflects the performance of the semantico-pragmatic reasoner independently of the other parts of the system.

Table 10

Synthesis of the results at the text level.

| | Training sample | | Validation sample | |
|-----------------------------|-----------------|-----------------|-------------------|-----------------|
| | Raw input | Corrected input | Raw input | Corrected input |
| Number of texts | 73 | | 87 | |
| Texts without errors | 67 (92%) | 72 (99%) | 70 (80%) | 81 (93%) |
| Texts with errors | 6 (8%) | 1 (1%) | 17 (20%) | 6 (7%) |

Text level

Finally Table 10 shows the results of the semantico-pragmatic reasoner at the text level. These results are the most significant ones in the evaluation of our approach. By “texts without errors”, we mean only those texts for which the system find exactly the expected primary as well as derived anomalies. All other texts are put in the class of texts with errors. It is clear that the system attains a good performance, as the rate of correct answers is 80% in the case of new texts (the validation sample). Moreover, if we are interested only in finding correctly the expected primary anomaly, the percentage of success goes up to 85%.

8.2.4. Robustness

If we compare the above figures with the percentage of texts in which there were errors in the semantic literals, we can notice a significant difference: while 60% of the validation texts contain at least one error in the semantic literals, we have errors in the primary or derived anomalies only for 20% of them. This result proves the robustness of the system. More precisely, among the 52 texts in the validation sample that contain one or several errors in their semantic literals, only 7 (13 %) lead to an erroneous determination of the primary or derived anomalies.

*8.3. Validation w.r.t. to a sample of human subjects**8.3.1. Setting of the experiment*

151 subjects, mainly first-year computer science students aged 18 to 23, have taken the test. They received a questionnaire on which one of the 10 selected reports (see Appendix A) was printed, so we have an average of 15.1 answers per report. After reading the report, they were asked to tick as many as they wanted among 8 to 10 potential causes that we proposed, the last one being “other cause, explain”. The next step was to circle one among these potential causes as the ‘main’ cause of the accident. The subjects were then asked the percentage of responsibility they assign to each of the drivers involved in the accident, and, on a scale from 1 (complete disagreement) to 7 (complete agreement), to which extent they agree to a given norm, chosen for its relevance to the circumstances of the accident.

The potential causes proposed include various factors, each being related to the accident, but sometimes in a very indirect way (e.g. *driver A took his car this very day*). For each text, one of the proposed causes expressed explicitly a duty (e.g. *vehicle B should have stopped*), but several other proposed causes referred implicitly to some duty (e.g. *vehicle B was driving too close from vehicle A*).

8.3.2. Main results

The task was meaningful for all subjects, even though, because of their age, some of them were new drivers or did not drive at all. A small amount of answers (less than 10%) look somewhat unreliable (because the cause circled as ‘main cause’ does not belong to the set of potential causes ticked, or the given percentages of responsibility are inconsistent with the causes selected). All the others are sensible, and they are all distinct: no two subjects have exactly the same opinion about the causes or the responsibilities of an accident.

Generally, approximately half of the potential causes proposed were ticked. The box “other cause, explain” was seldom used, and in that case, the subject expressed a very general statement (e.g. *people should drive more cautiously*). The only exception concerns Report B49 (see Appendix A): as the excessive speed of car#3 was not proposed as a potential cause although it clearly plays a role, several subjects used the box “other cause, explain” to mention it. This shows that the subjects did not merely play a kind of elimination game among the causes that we proposed, but tried to answer in a thoughtful way.

Another interesting feature of the answers is that most subjects did not hesitate to select as potential causes, and even as main cause, factors which were not mentioned in the report. This shows that the inferences performed during causal reasoning go far beyond the extraction of elements in the text.

Rather than a summary analysis of the results gathered on each of the 10 reports, we give a deeper analysis of two of them, one where the variations in the answers can be considered as superficial differences in the expression of an unanimous opinion; the other where the divergence is more fundamental.

Let us consider first Report A12:

I was driving my vehicle A on the right lane, reserved to vehicles heading straight ahead. Vehicle B was on the left lane, reserved to vehicles turning left (road-marking with arrows). Vehicle B cut in my vehicle A, hitting me on the left back.

Table 11

Reactions of the subjects to text A12 (potential causes 1 to 9: 0 = box not ticked, 1 = box ticked). The norm suggested here was the potential cause no. 5: *Driver B should have checked, before cutting in, that this maneuver was not dangerous.*

| Subject # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------------|----------|----------|----------|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Cause no. 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| no. 2 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| no. 3 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| no. 4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| no. 5 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| no. 6 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| no. 7 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| no. 8 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| no. 9 | 0 | 0 | 0 | Bad driver | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Main cause | 5 | 1 | 8 | 9 | 2 | 7 | 5 | 3 | 5 | 8 | 5 | 3 | 5 | 5 | 5 |
| % resp. A | 30 | 10 | 30 | 10 | 0 | 0 | 0 | 0 | 20 | 10 | 0 | 40 | 30 | 20 | 20 |
| % resp. B | 70 | 90 | 70 | 70 | 100 | 90 | 100 | 100 | 80 | 90 | 100 | 60 | 70 | 80 | 70 |
| Adhesion to norm | 7 | 6 | 5 | 5 | 7 | 7 | 7 | 7 | 7 | 5 | 7 | 7 | 7 | 6 | 7 |
| Age | 20 | 18 | 19 | 21 | 19 | 18 | 18 | 22 | 18 | 20 | 18 | 18 | 20 | 18 | 19 |

The potential causes proposed were:

1. Vehicle B cut in
2. A vehicle was blocking the left lane; to avoid it, vehicle B cut in
3. Vehicle B changed lane without putting his indicator on
4. Driver A did not avoid vehicle B when it came at his level
5. Driver B should have checked, before cutting in, that this maneuver was not dangerous
6. The presence of two parallel lanes
7. Vehicle B was driving too fast
8. Vehicle B did not respect the road marking
9. Other cause, explain: _____

The answers of the 15 subjects who received this text are given on Table 11.

Roughly half of the subjects took cause no. 5 as the 'main' cause of the accident; this is exactly the cause we expect: the accident is due to an agent who did not comply with his/her duty. Only one of the proposed cause suggested that driver A could have a share of responsibility (cause no. 4) and none of these subjects ticked it. Not surprisingly, they give a high percentage of responsibility to B (average 81%).

Five more subjects selected as 'main' causes different unfulfilled duties of B (even though they did not appear explicitly as duties, since the modal *should* is absent): causes nos. 3, 7 and 8.

Causes nos. 1, 2, 9 got a vote each: 1 is not an absolute violation of norm, but understood in context, it becomes so; 2 provides an explanation to driver B's behavior, but clearly the subject who selected no. 2 as a main cause does not take this explanation as an excuse, as he gives 100% responsibility to B. Finally, the subject giving 9 ("bad driver") as the main cause ticked all the potential causes incriminating B.

So B is unanimously considered as mainly or entirely responsible of the accident. Moreover, 12 to 14 of the 15 subjects, depending on how causes nos. 1 and 2 are considered, take as the 'main' cause of the accident the fact that an agent did not comply with his/her duties. Even if the duty in question is not always the one that our system (and that we) selected (namely driver B should have been *On_normal_lane* at time 3 and s/he was able to do so, but s/he failed to be on his/her lane at time 4), it is clearly related to it: if the agent respected his/her duty to check, before any maneuver, that it was not dangerous, s/he would have stayed in his/her normal lane. Thus, we believe that these results validate our work, the discrepancy observed resulting mainly from our choice, motivated by practical reasons, to express all causes in terms of a limited kernel of concepts.

Our second example is Report B6:

While I was moving off at the green light, I (driver A) hit the back of the vehicle that was ahead of me (vehicle B). I barely touched this vehicle, since only its bumper is slightly damaged. As for my vehicle, it suffered no damage.

The potential causes proposed were:

1. Driver A moved off when the light turned green, without taking into account the vehicle before him
2. Driver A should have waited until vehicle B proceeds
3. Driver A was too close to vehicle B

Table 12Reactions of the subjects to text B6. The suggested norm here was the potential cause no. 2: *Driver A should have waited until vehicle B proceeds.*

| Subject # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------------|----------|-----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Cause no. 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| no. 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| no. 3 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| no. 4 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | 1 | 1 | 0 | 1 | 1 | 1 |
| no. 5 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | | 0 | 0 | 0 | 0 | 1 | 1 |
| no. 6 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | 1 | 0 | 1 | 0 | 1 | 0 |
| no. 7 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| no. 8 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| no. 9 | 0 | 0 | | 1 | | 0 | 1 | 1 | 1 | 0 | 1 | 0 | | 0 | 1 |
| no. 10 | 0 | 0 | stress | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Main cause | 8 | | 2 | 2 | 8 | 8 | 3 | 2 | 9 | 1 | 2 | 1 | 8 | 8 | 4 |
| % resp. A | 80 | 100 | 50 | 70 | 100 | 0 | 80 | 0 | 50 | 80 | 80 | 80 | 80 | 60 | 40 |
| % resp. B | 20 | 0 | 50 | 30 | 0 | 100 | 60 | 100 | 50 | 20 | 20 | 20 | 20 | 40 | 60 |
| Adhesion to norm | 6 | 7 | 7 | 6 | 7 | 1 | 4 | 3 | 5 | 7 | 7 | 7 | 6 | 6 | 7 |
| Age | 21 | 20 | 24 | 21 | 19 | 23 | 19 | 18 | 20 | 20 | 20 | 20 | 18 | 18 | 19 |

4. Vehicle B did not move off when the light turned green
5. The light turned green
6. Vehicle B was preceding vehicle A
7. Driver B did not notice that the light turned green
8. Driver A paid no attention to the vehicle before him
9. No cause; no damage (barely touched)
10. Other cause, explain: _____

The answers of the 15 subjects who received this text are given on Table 12.

The duty appears explicitly only in cause no. 2; this cause has been ticked by 10 of the 15 subjects, but only 4 of them consider it as the 'main' cause. However causes nos. 1, 3, 4 and 8 correspond to implicit duties which have not been fulfilled. Altogether, 13 out of the 15 subjects selected as main cause an unfulfilled duty.

But looking now at the responsibilities, it is clear that only 10 subjects consider, as we do, that the accident is due to the inattention of A. As a matter of fact, our system finds as primary anomaly, i.e. as cause of the accident the fact that at time 1, driver A had the duty not to move off and was able not to move off; however, at time 2, A moved off. Nonetheless, 3 subjects consider B as more, or exclusively responsible of the accident, and the 2 last subjects share the responsibility 50–50.

What is also noticeable is the fact that nothing in the text evokes causes nos. 3, 4, 7, 8. Despite of this, 14 out of the 15 have selected at least one of them as a potential cause of the accident. Causes nos. 5 and 6 are distractors: of course, one can argue that if the light had turned green later, A might have paid attention and the accident would not have occurred; similarly if B was not preceding A, A would not have hit B. So there is a causal relation in the counterfactual sense (this can explain why 9 out of the 15 subjects ticked either one or the other of these causes), but if someone, asked about the cause of this accident, answered: "the light turned green", her seriousness would be under question!

A close examination of the results gathered for the 8 other reports of the experiment reveals various intermediaries between the quasi-unanimity of A12 and the significant variations observed for B6. The ascription of causes to abnormal factors is massively validated in all cases.

9. Conclusion and perspectives

The system that has been presented in this paper has a number of interesting features, as well as several weaknesses.

On the positive side, it accepts genuine car crash reports that had not been seen beforehand; it is robust and quick; and four times out of five, it gives as causes of the accident exactly the same elements as we do. But as the experiment reported in Section 8.3 shows, this result is not as significant as what we thought initially, since our causal judgments are not as widely agreed with as we believed. But in most cases the answers of the subjects differ slightly from our own answers, and when the difference is significant, it seems to be explained by the fact that a number of subjects give higher importance than we do to some norms, and not in their using something else than the notion of norm to ascribe the cause of the accident. So in any case, we consider that the postulate relating cause with norm has been validated.

Our initial objective could thus be broadened: instead of getting the (most frequently stated?) causes of the accident, we could try and find all elements that, say, more than 10 percent of the subjects consider as causes. This kind of investigations would be closer to Cognitive Modeling than to AI.

For what concerns NL semantics, we make no use of the regular (compositional) tools. This work shows the feasibility of an inferential approach to NL requiring neither a correct nor a complete parser, and based on norms rather than on the preservation of truth.

As a perspective for future work, we believe that following questions should have priority:

- How portable is the system? If we switch from the domain of road accidents to other kinds of reports where causality plays an important role, would the postulates stated at the beginning of this paper still remain valid?
- More specifically, if it is agreed that collecting the norms of a given domain is of prominent importance, that most of them are kept implicit and are not to be found in the handbooks or in similar items concerning the domain, is the approach adopted here, namely try to reproduce the causal reasoning of ‘experts’, a viable method to gather the norms they have in mind?
- How scalable is it? Far from the presumptuousness of works attempting to incorporate a significant part of mankind’s encyclopedic knowledge, would it be nonetheless sensible to increase gradually the domain covered?
- In any case, whether the goal is to improve the performance on the same domain, to change, or to extend the domain, should we rework completely the Natural Language analyzers, and incorporate general-purpose resources, or would it add too much noise, and is part of our current success precisely due to the fact that we tailored the analyzers to our exact needs?

Answering these questions requires further work. But the reward will be both a step forward in the vexed question of what “cause” really means for us, and a better understanding of the reasons why, after half a century of efforts and an incredible progress in the speed and storage capacity of the computers, Natural Language Understanding is still in its infancy.

Acknowledgements

We express our gratefulness to Françoise Gayral, François Lévy, Catherine Recanati for stimulating discussions and suggestions, and to the anonymous referees for helpful criticisms on the first version of this paper. Philippe Chassy and Denis Hilton at Université Toulouse Le Mirail provided an invaluable help for the design and execution of the experiment described in Section 8.3. and in analyzing the data collected. We also thank Denis Hilton for his helpful suggestions in improving the English of the paper. The work presented here has been partially supported by the MICRAC project, funded by ANR (Agence Nationale de la Recherche).

Appendix A. A sample of the corpus

Report D1. *En sortant de mon domicile, j’ai dû braquer pour éviter un véhicule qui arrivait à vive allure, et j’ai heurté une des bornes en ciment qui longent le trottoir.*

(lit. As I was driving out of my home, I was forced to swing to avoid a vehicle arriving at high speed, and I bumped into one of the concrete markers that border the sidewalk.)

Report D37. *Je roulais sur la route Nom_De_Voie1 direction Nom_De_Lieu. Le véhicule B venait d’une voie de circulation par la gauche. Le véhicule B n’a pas respecté le stop et m’a percuté à l’avant gauche. En plus des dégâts constatés, un pneu de ma voiture a été détérioré.*

(lit. I was driving on road#1, in the direction of location#1. Vehicle B was coming from a traffic way at the left. Vehicle B did not respect the stop sign and smashed into me in the front left. In addition to the recorded damages, a tire of my car has been deteriorated.)

Report A4. *Véhicule B venant de ma gauche, je me trouve dans le carrefour, à faible vitesse environ 40 km/h, quand le véhicule B percute mon véhicule, et me refuse la priorité à droite. Le premier choc atteint mon aile arrière gauche, sous le choc, et à cause de la chaussée glissante, mon véhicule dérape, et percute la protection métallique d’un arbre, d’où un second choc frontal.*

(lit. Vehicle B coming from my left, I am at the crossroads, at low speed about 25 mph, when vehicle B smashes into my vehicle and refuses to yield, although I have priority. The first shock hits my left back fender; as the roadway was slippery, under the crash my vehicle skids, and hits the metal protection of a tree, hence a second head-on shock.)

Report B6. *Alors que je redémarrais au feu vert, j’ai heurté l’arrière du véhicule qui me précédait. J’ai à peine touché ce véhicule puisque seul son pare-chocs est légèrement abîmé. Mon véhicule n’ayant lui, subi aucun dégât.*

(lit. While I was moving off at the green light, I hit the back of the vehicle that was ahead of me. I have barely touched this vehicle, since only its bumper is slightly damaged. As for my vehicle, it suffered no damage.)

Report D2. *Je venais de quitter mon stationnement et j’allais sortir du parking. J’étais à l’arrêt, car j’attendais de pouvoir m’engager sur l’avenue Nom_De_Voie. C’est alors que le véhicule A, en reculant, m’a heurté. Je n’ai pas pu m’avancer, à cause de la circulation sur l’avenue, ni reculer à cause des voitures derrière moi, de manière à éviter le choc.*

(lit. I had just left my parking place and I was about to leave the parking lot. I was stationary, because I was waiting for an opportunity to step out onto avenue#1. At this very moment, vehicle A, moving backwards, bumped into me. I could not move forward, because of the traffic on the avenue, nor could I move backwards since there were cars behind me, in order to avoid the crash.)

Report B16. *Le véhicule A roulait lentement en ligne droite tout près de son domicile ; le véhicule était en stationnement sur une bande réservée à cet effet et la conductrice a ouvert la portière gauche avant, provoquant, de ce fait, un choc : le rétroviseur du véhicule A (côté droit) fut brisé au moment du passage du véhicule A.*

(lit. Vehicle A was driving slowly in straight line, very close to its home; the vehicle was parked on a lane reserved for that purpose and the driver opened the left front door, causing for this reason a shock: vehicle A's side-view mirror (right side) broke at the moment when vehicle A came past.)

Report A12. *Je circulais à bord de mon véhicule A sur la file de droite réservée aux véhicules allant tout droit. Le véhicule B circulait sur la voie de gauche réservée aux véhicules allant à gauche (marquage au sol par des flèches). Celui-ci s'est rabattu sur mon véhicule A me heurtant à l'arrière gauche.*

(lit. I was driving on board my vehicle A on the right lane reserved to vehicles going straight ahead. Vehicle B was moving on the left lane reserved to vehicles turning left (road-marking with arrows). The latter cut in my vehicle A, hitting me on the left back.)

Report C3. *Ma voiture (véhicule B) était stationnée sur le parking où je travaille. J'étais sortie de mon véhicule et me tenais sur le parking, lorsqu'une voiture, garée derrière moi plus haut, a dévalé le parking, sans conducteur à l'intérieur et a heurté ma voiture sur le côté gauche en la poussant.*

(lit. My car (vehicle B) was parked on the parking lot of where I work. I was out of my vehicle and was standing on the parking lot, when a car, parked behind me at a higher place, hurtled down the parking lot, with no driver inside, and bumped into the left side of my car and pushed it.)

Report C10. *La voiture B a glissé et s'est retrouvée en travers de la route (perpendiculaire à la route). J'arrivais derrière et, à cause du verglas, je n'ai pas pu m'arrêter. Mon véhicule a percuté le véhicule B, et mon véhicule a été projeté dans le fossé.*

(lit. Vehicle B slipped and ended up crosswise the road (perpendicular to the road). I was arriving behind and, because of the ice, I have been unable to stop. My vehicle bumped into vehicle B, and my vehicle has been hurled into the ditch.)

Report B49. *Arrêtée à un feu rouge avec le pied sur le frein, frein à main levé, j'ai reçu un choc brutal à l'arrière provenant d'une Nom_De_Véhicule1 immatriculée Numéro_D_Immatriculation1. J'ai moi-même alors été projeté contre la voiture me précédant: Nom_De_Véhicule2 immatriculée Numéro_D_Immatriculation2. Il pleuvait à verse et la chaussée était glissante. Le premier choc venait en fait d'une Nom_De_Véhicule3 immatriculée Numéro_D_Immatriculation3 qui a initialement touché la Nom_De_Véhicule1.*

(lit. Stopped at a red light, with the foot on the brake pedal, handbrake up, I got a violent shock at the back, due to a brand#1 car license plate no.#1. I have myself been hurled onto the car ahead of me: a brand#2 car license plate no.#2. It was pouring down and the roadway was slippery. The first shock was in fact due to a brand#3 car license plate no.#3 that initially hit the brand#1.)

Appendix B. List of the semantic predicates

A, B: an agent (by metonymy, it can refer to either a vehicle or a person).

O: an object (it can be an agent or another obstacle e.g. a wall, a (sign)post ...).

C: a color.

P: a position (back, front, ...).

D: a direction (left, right).

X: a cause of loss of control (icy patch, bits of gravel, ...).

L: a location (parking, garage, ...).

R: a circulation area (street, lane, road, ...).

| Predicate | Meaning |
|-------------------------------------|---|
| holds (Is_approaching_Xroads, A, T) | A is approaching a crossroads |
| holds (Access_adj_lane, A, T) | A accesses the adjacent lane |
| holds (Stop, A, T) | A is stopped |
| holds (Is_arriving_on_file, A, T) | A is arriving on a file |
| holds (Has_driver, A, T) | A is with driver |
| holds (Give_way, A, T) | A is in the presence of a give way sign |
| holds (Control, A, T) | A is under control |
| holds (Is_overtaking, A, T) | A is overtaking |
| holds (Is_moving_off, A, T) | A moves off |
| holds (Is_wrong_way, A, T) | A takes the wrong way |
| holds (Mistaken_command, A, T) | A makes a command mistake |
| holds (Is_braking, A, T) | A brakes |
| holds (Brake_release, A, T) | The brake of A released |

| Predicate | Meaning |
|--|--|
| holds (Is_slipping, A, T) | A slips |
| holds (Police, A, T) | A is stopped by the police |
| holds (Door_open, A, T) | A door of vehicle A is open |
| holds (Pedestrian, A, T) | A is in the presence of a pedestrian |
| holds (Is_moving_backwards, A, T) | A is going backwards |
| holds (Steer_release, A, T) | A releases the steering wheel |
| holds (Drives_app, A, T) | A drives at an appropriate speed |
| holds (Is_moving, A, T) | A is moving |
| holds (Signal, A, T) | A signals something |
| holds (Parked, A, T) | A is parked |
| holds (On_normal_lane, A, T) | A drives on the good (normal) lane |
| holds (Stop, A, T) | A is in the presence of a “stop” sign |
| holds (Is_falling, A, T) | The driver of A is ejected outside the vehicle |
| holds (Bend, A, T) | A arrive on a road bend |
| holds (Change_dir, A, T) | A change slightly its direction |
| holds (combine(At_right_of, B), A, T) | B is situated at the right of A |
| holds (combine(Cause_not_control, X), A, T) | X is a potential cause of control loss for A |
| holds (combine(Shock, O), A, T) | There is a shock between A and O (the symmetric holds too) |
| holds (combine(Is_cutting_road, B), A, T) | A cuts the road of vehicle B |
| holds (combine(Avoid, O), A, T) | A avoid the obstacle O |
| holds (combine(Red_light, C), A, T) | There is a red light for A |
| holds (combine(Is_hitting, O), A, T) | A bumps on the obstacle O |
| holds (combine(Same_file, B), A, T) | A and B are in the same file |
| holds (combine(Same_road, B), A, T) | A and B drive in the same road |
| holds (combine(Same_direction, B), A, T) | A and B drive in the same direction |
| holds (combine(Same_line, B), A, T) | A and B are in the same line |
| holds (combine(Is_disturbing, B), A, T) | A disturbs B |
| holds (combine(Pos_shock, P), A, T) | The position of the shock of A is P |
| holds (combine(Precede, B), A, T) | A precedes B in a file |
| holds (combine(Has_priority, B), A, T) | B has priority over B |
| holds (combine(Opposite_direction, B), A, T) | A and B run in opposite directions |
| holds (combine(Is_leaving, L), A, T) | A leaves place L |
| holds (combine(Follows, B), A, T) | A follows B in a file |
| holds (combine(Is_turning, D), A, T) | A turns in the direction D |
| holds (combine(Is_crossing, R), A, T) | A crosses the road |
| holds (combine(Change_dir, D), A, T) | A changes its direction in the direction D |
| Fog ^a | There is fog |
| Slow_down | There is a general deceleration |
| Slippery_road | The road is slippery |

^a The three last predicates of this list have arity zero, as they are not linked to any agent, and they are supposed to remain true during the whole scene.

Appendix C. Rules used for the running example

Rule (11) works for expressions of the form “the vehicle skidded... on/because of... the glaze/gravels.” and allows the inference of the predicate expressing a loss of control due to an identified cause.

```
(11) type (A, vehicle) ∧ type (X, cause_loss_control) ∧ prep (T) ∧
    sem_rep (V, is_slipping) ∧ holds_I (V, A, W) ∧
    holds_I (combine(combine(T, V), X), A, temp_ref(T)) →
    holds (combine(cause_not_control, X), A, temp_ref(T))
```

Rule (12) infers the presence of the vehicle in a bend.

```
(12) type (A, vehicle) ∧ sem_rep (X, bend) ∧ verbe (P) ∧ prep (T) ∧
    holds_I (combine(combine(T, P), X), A, temp_ref(T)) → holds (bend, A, temp_ref(T))
```

Given the static predicate type (gravels, cause_loss_control) (*gravels are known to potentially cause a loss of control*), these two rules infer in our running example the atemporal semantic literal: holds (combine(cause_not_control, gravels), author, temp_ref(sur₂)) and: holds (bend, author, temp_ref(dans)).

```
(13) type (A, vehicle) ∧ type (X, cause_loss_control) ∧ prep (T) ∧
    sem_rep (V, is_slipping) ∧ holds_I (V, A, W) ∧
    holds_I (combine(combine(T, V), X), A, temp_ref(T)) → prec (temp_ref(T), W)
```

```
(14) type (A, vehicle) ∧ sem_rep (X, bend) ∧ verbe (P) ∧ prep (T) ∧
    holds_I (combine(combine(T, P), X), A, temp_ref(T)) →
    prec (temp_ref(T), temp_ref(P))
```

Rule (13) simply expresses that if some event appears which causes a loss of control, then this event precedes naturally the loss of the control itself; it yields: `prec (temp_ref(sur2), temp_ref(déraper))`.

The second temporal relation inferred in our example expresses that the loss of control happened after the access to the road bend by the driver: `prec(temp_ref(dans), temp_ref(déraper))`.

This relation which agrees, in our text, with the narrative order²¹ is obtained thanks to rule (14).

The semantic literals obtained by the linguistic reasoner for report B2.B are reproduced here:

```
holds (is_moving, author, 1), holds (bend, author, 2),
holds (combine(cause_not_control, gravels), author, 2),
holds (is_slipping, author, 3), holds (is_falling, author, 4),
holds (is_moving, veh_A, 5), ¬holds (Stop, veh_A, 6),
holds (combine(is_hitting, author), veh_A, 7)
```

The semantic inference rules that apply to the text are:

(19) `holds (combine(cause_not_control, gravels), A, T) ∧ holds (bend, A, T) → predictable (combine(cause_not_control, gravels), A, T)`²²

It states that “*the presence of gravels in a road bend represents a predictable cause of loss of control*” since when entering a bend, one should anticipate the presence of unseen obstacles. The application of this rule allows to infer ($A = \text{author}$, $T = 2$): `predictable (combine(cause_not_control, gravels), author, 2)`.

Rules (20) and (21) impose initial conditions such as: “*at time 0, each vehicle is under control and with a driver*”.

(20) `agent(A) → holds (Control, A, 0)`

(21) `agent(A) → holds (Has_driver, A, 0)`

By applying these rules, we obtain ($A = \text{author}$, then $A = \text{veh_A}$): `holds (Control, author, 0), holds (Control, veh_A, 0) and holds (Has_driver, author, 0), holds (Has_driver, veh_A, 0)`.

Rule (22) expresses the fact that “*a driver falling from his/her vehicle is an unpredictable event*”.

(22) `holds (is_falling, A, T) → ¬predictable (is_falling, A, T)`

This rule provides us with ($A = \text{author}$, $T = 4$): `¬predictable (is_falling, author, 4)`.

Rule (23) tells that “*if the driver of a vehicle falls at time T, then his/her vehicle will be without driver at time T + 1*”.

(23) `holds (is_falling, A, T) → ¬holds (Has_driver, A, T+1)`

We get by this rule ($A = \text{author}$, $T = 4$): `¬holds (Has_driver, author, 5)`.

The next rule (24) allows to infer that the agent *author* lost the control of his/her vehicle at time 3:

(24) `holds (is_slipping, A, T) → ¬holds (Control, A, T)`

Hence ($A = \text{author}$, $T = 3$): `¬holds (Control, A, 3)`.

The forward persistence of the predicate *Control* allows to infer that agent *author* has control at times 1 and 2 while agent *veh_A* has control in every temporal state of the scene. In the same way, by applying the forward persistence of the predicate *Has_driver*, we can infer that the vehicle of agent *author* is with its driver at times 1, 2, 3 and 4 while this predicate holds for agent *veh_A* at all the states of the scene:

```
holds (Control, author, T), for T in [1, 2]
holds (Control, veh_A, T), for T in [1, 7]
holds (Has_driver, author, T), for T in [1, 4]
holds (Has_driver, veh_A, T), for T in [1, 7]
```

The forward persistence of the negation of *Control* and *Has_driver* yields:

```
¬holds (Control, author, T), for T in [3, 7]
¬holds (Has_driver, author, T), for T in [5, 7]
```

²¹ In other cases, this rule infers a temporal precedence in opposition to the narrative order; this would have been the case if our example read: “The motorbike skidded in the curve”, instead of: “In the curve, the motorbike skidded” since obviously in both cases, the bike was in the curve before it skidded.

²² In fact, this rule blocks a default according to which gravels belong in general to unpredictable causes of loss of control.

Rules (25) and (26) state the obvious facts that if a vehicle A bumps on a vehicle B, then there is a shock between them (25) and that the predicate shock is symmetric:

(25) holds (combine(is_hitting, A), B, T) → holds (combine(shock, A), B, T)

(26) holds (combine(shock, A), B, T) → holds (combine(shock, B), A, T)

Applying these rules, we obtain (A = author, B = veh_A, T = 7): holds (combine(shock, author), veh_A, 7) and holds (combine(shock, veh_A), author, 7).

According to rule (27) "if there is a shock between A and B at time T, then we can deduce by default that A did represent an obstacle for B at time T – 1, except if we know that B has avoided A at time T – 1 or that B is not a vehicle at all"

(27) holds (combine(shock, A), B, T) : holds (combine(obstacle, A), B, T-1)
[¬holds (combine(avoid, A), B, T), agent(B)]

By this rule, we can infer (A = author, B = veh_A, T = 7 then A = veh_A, B = author, T = 7): holds (combine(obstacle, author), veh_A, 6) and holds (combine(obstacle, veh_A), author, 6).

Rule (28) expresses that: "if X is an obstacle for A at time T and X is a vehicle which is not under control at this state, then X is an unpredictable obstacle for A at time T".

(28) ¬holds (Control, X, T) ∧ holds (combine(obstacle, X), A, T) →
¬predictable (combine(obstacle, X), A, T)

This rule infers (X = author, A = veh_A, T = 6): ¬predictable (combine(obstacle, author), veh_A, 6).

Finally, rule (29) states that: "if at time T, an agent A is in a road bend and at the next temporal state, T + 1, A is not under control, the default assumption is that A was not driving at an appropriate speed at time T"

(29) holds (bend, A, T) ∧ ¬holds (Control, A, T+1) : ¬holds (Drives_app, A, T)

We obtain thus: (A = author, T = 2): ¬holds (Drives_app, author, 2).

The first relevant rule for our example concerning duties is (30). It states that "in every state, a driver must keep the control of his/her vehicle unless control is already lost at this state".

(30) : duty (Control, A, T) [holds (Control, A, T)]

This default is applicable and yields e.g. (A = author, T = 2): duty (Control, author, 2).

Analogously, the rule (31) expresses that "in every state, a vehicle must be with its driver except if it is stopped, or if it is already without driver".

(31) : duty (Has_driver, A, T) [holds (Has_driver, A, T), ¬holds (Stop, A, T)]

This default is applied (A = author, T = 4) and yields: duty (Has_driver, author, 4).

According to the rule (32): "one must avoid any obstacle appearing in the road".

(32) holds (combine(obstacle, X), A, T) → duty (combine(avoid, X), A, T)

The following literals obtain (X = author, A = veh_A, T = 6 then X = veh_A, A = author, T = 6): duty (combine(avoid, author), veh_A, 6) and duty (combine(avoid, veh_A), author, 6).

Rule (33) translates by default the duty to avoid an obstacle into a duty to stop. Many exceptions block this default. They are listed in the justification part of the default.

(33) duty (combine(avoid, B), A, T) ∧ holds (combine(shock, B), A, T+1)
: duty (Stop, A, T) [¬duty (Drives_app, A, T), ¬holds (Stop, A, T),
¬holds (combine(same_file, A), B, T), ¬duty (not(Is_moving_backwards), A, T-1),
¬duty (not(Is_moving_off), A, T-1), predictable (combine(obstacle, B), A, T)]

We obtain by applying this rule (B = veh_A, A = author, T = 6): duty (Stop, author, 6).

However, this same default is blocked for the agent veh_A. In fact, the agent author is an unpredictable obstacle for the agent veh_A at state 6 (see (28) above), and this is the negation of one of the justifications of the default.

Rule (46) performs a propagation of duties: if a given duty is not respected because of a predictable event, and if avoiding this event depends on respecting some precondition, then we infer the duty to respect this precondition:

(46) $\text{effect}(F) \wedge \text{effect}(F_1) \wedge \text{event}(\text{Evt}) \wedge \text{duty}(F, A, T+1) \wedge \text{pot_cause}(\text{Evt}, F_1) \wedge$
 $\text{incompatible}(F, F_1) \wedge \text{holds}(\text{Evt}, A, T+1) \wedge \text{predictable}(\text{Evt}, A, T+1) \wedge$
 $\text{precond_avoid_event}(\text{Evt}, F_1, P) \rightarrow \text{duty}(P, A, T)$

This rule allows to deduce the duty at state 1 of the agent *author* to drive slowly ($T = 1, A = \text{author}, P = \text{Drives_app}, F = \text{Control}, F_1 = \text{not}(\text{Control}), \text{Evt} = \text{combine}(\text{cause_not_control}, \text{gravels})$): $\text{duty}(\text{Drives_app}, \text{author}, 1)$.

This new duty leads us to calculate the capacity of the agent *author* to drive fairly slowly at time 1. Knowing that the action of braking is a potential cause of the fact of driving fairly slowly and that no exception can block the default (36) concerning the default availability of actions, we can deduce: $\text{available}(\text{brake}, \text{Drives_app}, \text{author}, 1)$.

The application of the rule (40) once again, allows thus to obtain: $\text{is_able_to}(\text{Drives_app}, \text{author}, 1)$.

Let us sum up now the anomalies we can detect. From the literals: $\text{duty}(\text{Control}, \text{author}, 2)$, $\neg \text{is_able_to}(\text{Control}, \text{author}, 2)$, $\neg \text{holds}(\text{Control}, \text{author}, 3)$, rule (45) finds:

$\text{Derived_anomaly}(\text{Control}, \text{author}, 3)$

i.e. “the fact that the agent *author* lost control at time 3 is a derived anomaly”. Analogously, we detect two other derived anomalies. The first one concerns the fact that the author’s vehicle was without driver at time 5 and the fact that the agent *author* did not stop at time 7.

$\text{Derived_anomaly}(\text{Has_driver}, \text{author}, 5)$
 $\text{Derived_anomaly}(\text{Stop}, \text{author}, 7)$

Finally, by rule (43) with premises: $\text{duty}(\text{Drives_app}, \text{author}, 1)$, $\text{is_able_to}(\text{Drives_app}, \text{author}, 1)$, $\neg \text{holds}(\text{Drives_app}, \text{author}, 2)$, we get the result:

$\text{Primary_anomaly}(\text{Drives_app}, \text{author}, 1)$

References

- [1] V. Akman, S.T. Erdogan, J. Lee, V. Lifschitz, H. Turner, Representing the zoo world and the traffic world in the language of the causal calculator, *Artificial Intelligence* 153 (1–2) (2004) 106–140.
- [2] J. Allen, H.A. Kautz, A model of naive temporal reasoning, in: J. Hobbs, R.C. Moore (Eds.), *Formal Theories of the Commonsense World*, in: *Ablex Series in Artificial Intelligence*, 1985, pp. 251–268.
- [3] C. Anger, M. Gebser, T. Linke, A. Neumann, T. Schaub, The *nomore++* system, in: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), *Proceedings of the 8th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR’05)*, Diamante, Italy, in: *LNAI*, vol. 3662, Springer-Verlag, Berlin, 2005, pp. 422–426.
- [4] S. Benferhat, et al., A comparative study of six formal models of causal ascription, in: S. Greco, T. Lukasiewicz (Eds.), *Proceedings of the 2nd International Conference on Scalable Uncertainty Management*, Naples, Italy, in: *LNAI*, vol. 5291, Springer-Verlag, Berlin, 2008, pp. 47–62.
- [5] B. Bennett, A.P. Galton, A unifying semantics for time and events, *Artificial Intelligence* 153 (1–2) (2004) 13–48.
- [6] N. Bidoit, C. Froidevaux, General logical databases and programs: Default logic, Semantics and Stratification, *Information and Computation* 91 (1) (1991) 15–54.
- [7] D.G. Bobrow, Natural Language input for a computer problem solving system, in: M. Minsky (Ed.), *Semantic Information Processing*, The MIT Press, 1967, pp. 133–215.
- [8] M. Boman, Norms in artificial decision-making, *Artificial Intelligence and Law* 7 (1) (1999) 17–35.
- [9] S. Boutouhami, D. Kayser, Vers la construction de descriptions argumentées d’un accident de la route: analyse de diverses stratégies argumentatives, *Corela* 6 (1) (2008), available online at <http://edel.univ-poitiers.fr/corela/document.php?id=1887>.
- [10] J.D. Bransford, J.J. Franks, The abstraction of linguistic ideas, *Cognitive Psychology* 2 (1971) 331–350.
- [11] G. Brewka, Cumulative default logic: in defense of nonmonotonic inference rules, *Artificial Intelligence* 50 (2) (1991) 183–205.
- [12] K. Chan, W. Lam, Extracting causation knowledge from natural language texts, *International Journal of Intelligent Systems* 20 (2005) 327–358.
- [13] G. Chierchia, S. McConnell-Ginet, *Meaning and Grammar. An Introduction to Semantics*, The MIT Press, 1990.
- [14] A. Colmerauer, Les systèmes-Q, un formalisme pour analyser et synthétiser des phrases sur ordinateur, publication n°43, département d’Informatique de l’Université de Montréal 1970 (reprinted in *TAL* 33 (1–2) (1992) 105–148).
- [15] J. De Kleer, J.S. Brown, A qualitative physics based on confluence, *Artificial Intelligence* 24 (1–3) (1986) 7–83.
- [16] J.P. Delgrande, An approach to default reasoning based on a first-order conditional logic, *Artificial Intelligence* 36 (1) (1988) 63–90.
- [17] F. Dignum, D. Kinny, L. Sonenberg, From desires, obligations and norms to goals, *Cognitive Science Quarterly* 2 (3–4) (2002) 405–427.
- [18] D. Dubois, H. Prade, Modeling the role of (ab)normality in the ascription of causality judgments by agents, in: *Proceedings of IJCAI-05 Workshop on Nonmonotonic Reasoning, Action, and Change*, Edinburgh, Scotland, 2005, pp. 22–27.
- [19] (Sous la direction de) P. Enjalbert, *Sémantique et traitement automatique du langage naturel*, Hermès Science Publications, Paris, 2005.
- [20] D.W. Etherington, Formalizing nonmonotonic reasoning systems, *Artificial Intelligence* 31 (1) (1987) 41–85.
- [21] D. Garcia, Analyse automatique des textes pour l’aide à l’organisation des phénomènes, des activités et des actions, Réalisation du système informatique COATIS, Thèse de l’Université Paris-IV Sorbonne, 1998.
- [22] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, *New Generation Computing* 9 (3–4) (1991) 363–385.
- [23] M.L. Ginsberg, D.E. Smith, Reasoning about action II: the qualification problem, *Artificial Intelligence* 35 (3) (1988) 311–342.
- [24] L. Giordano, C. Schwind, Conditional logic of actions and causation, *Artificial Intelligence* 157 (1–2) (2004) 239–279.
- [25] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, H. Turner, Nonmonotonic causal theories, *Artificial Intelligence* 153 (1–2) (2004) 49–104.
- [26] B. Green, A.K. Wolf, C. Chomsky, K. Laughery, *BASEBALL: An automatic question answerer*, in: E.A. Feigenbaum, J. Feldman (Eds.), *Computers and Thought*, The MIT Press, 1963, pp. 207–216.
- [27] J. Groenendijk, M. Stokhof, Dynamic predicate logic, *Linguistics and Philosophy* 14 (1991) 39–100.
- [28] G. Gross, Verbes supports et conjugaison nominale, *Revue d’Etudes Francophones* 9 (1999) 70–92.

- [29] Y. Hagmayer, D. Hilton, Causal reasoning in practice – how causal assumptions affect explanations, inferences and decision making, in: S. Vosniadou, D. Kayser, A. Protopapas (Eds.), *Proceedings of EuroCogSci. 07, Delphi, Greece, Lawrence Erlbaum Ass., 2007*, p. 40.
- [30] J.Y. Halpern, J. Pearl, Causes and explanations: a structural-model approach. Part I: Causes, *British Journal for Philosophy of Science* 56 (4) (2005) 843–887 and Part II: Explanations, *British Journal for Philosophy of Science* 56 (4) (2005) 889–911.
- [31] D.J. Hilton, J.L. McClure, B.R. Slugoski, The course of events: counterfactuals, causal sequences and explanation, in: D. Mandel, D.J. Hilton, P. Catellani (Eds.), *The Psychology of Counterfactual Thinking*, The Psychology Press, London, 2005, pp. 44–73.
- [32] C. Ioannides, S. Vosniadou, The changing meanings of force, *Cognitive Science Quarterly* 2 (1) (2002) 5–61.
- [33] Y. Iwasaki, H.A. Simon, Causality in device behavior, *Artificial Intelligence* 29 (1) (1986) 3–32.
- [34] R. Johanson, A. Berglund, M. Danielsson, P. Nagues, Automatic text-to-scene conversion in the traffic accident domain, in: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, 2005, pp. 1073–1078.
- [35] H. Kamp, U. Reyle, *From Discourse to Logic*, Kluwer, Dordrecht, 1993.
- [36] D. Kayser, A. Mokhtari, Time in a causal theory, *Annals of Mathematics and Artificial Intelligence* 22 (1–2) (1998) 117–138.
- [37] M. Kistler, *Causalité et lois de la nature*, Vrin (coll.Mathesis), Paris, 1999 (English version: *Causation and Laws of Nature*, Routledge, London, 2006).
- [38] S. Kraus, D. Lehmann, M. Magidor, Non-monotonic reasoning, preferential models, and cumulative logics, *Artificial Intelligence* 44 (1–2) (1990) 167–207.
- [39] D. Lehmann, M. Magidor, What does a conditional knowledge base entail? *Artificial Intelligence* 55 (1) (1992) 1–60.
- [40] W. Lehnert, *The Process of Question-Answering*, Lawrence Erlbaum Ass., Hillsdale, 1978.
- [41] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, F. Scarcello, The DLV system for knowledge representation and reasoning, *ACM Transactions on Computational Logic* 7 (3) (2006) 499–562.
- [42] D. Lewis, *Philosophical Papers*, vol. 2, Oxford University Press, Oxford, 1986.
- [43] Y. Lierler, M. Maratea, Cmodels-2: Sat-based answer set solver enhanced to non-tight programs, in: V. Lifshitz, I. Niemelä (Eds.), *Proceedings of the 7th International Conference on Logic Programming and NonMonotonic Reasoning (LPNMR'04)*, Fort Lauderdale, USA, in: *LNAI*, vol. 2923, Springer-Verlag, Berlin, 2004, pp. 346–350.
- [44] V. Lifshitz, Answer sets in general nonmonotonic reasoning, in: *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR-92)*, Morgan-Kaufmann, Cambridge, USA, 1992, pp. 603–614.
- [45] J.L. Mackie, *The Cement of the Universe: A Study of Causation*, Oxford University Press, 1974.
- [46] J. McCarthy, P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Melzer, D. Michie (Eds.), *Machine Intelligence*, vol. 4, Edinburgh University Press, 1969, pp. 463–502.
- [47] D.V. McDermott, A temporal logic for reasoning about processes and plans, *Cognitive Science* 6 (1982) 101–155.
- [48] P. McNamara, H. Prakken (Eds.), *Norms, Logics and Information Systems*, New Studies in Deontic Logic and Computer Science, *Frontiers in Artificial Intelligence and Applications*, vol. 49, IOS Press, 1999.
- [49] M. Minsky, A framework for representing knowledge, A.I. Memo 306, M.I.T., 1974 (reprinted in: P.H. Winston (Ed.), *The Psychology of Computer Vision*, McGraw Hill, 1975, pp. 211–277).
- [50] R. Mitkov, *Anaphora Resolution*, Pearson Education, 2002.
- [51] R. Montague, in: R. Thomason (Ed.), *Formal Philosophy*, Yale University Press, 1974.
- [52] A. Nazarenko, *La cause et son expression en Français*, Ophrys, Paris, 2000.
- [53] F. Nouioua, *Extraction et Utilisation des normes pour un raisonnement causal dans un corpus textuel*, Thèse de l'Université Paris XIII, 2007.
- [54] F. Nouioua, P. Nicolas, Using answer set programming in an inference-based approach to natural language semantics, in: *Proceedings of the 5th International Workshop on Inference in Computational Semantics (ICoS-5)*, Buxton, England, 2006, pp. 77–86.
- [55] J. Pearl, *Causality. Models, Reasoning, and Inference*, Cambridge University Press, 2000.
- [56] R. Reiter, A logic for default reasoning, *Artificial Intelligence* 13 (1–2) (1980) 81–132.
- [57] R. Reiter, G. Criscuolo, On interacting defaults, in: *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, Vancouver, Canada, pp. 270–276.
- [58] R.C. Schank, R.P. Abelson, *Scripts, Plans, Goals and Understanding*, Lawrence Erlbaum Ass., Hillsdale, 1977.
- [59] T. Syrjänen, I. Niemelä, The Smodels systems, in: *Proceedings of the 6th International Conference on Logic Programming and NonMonotonic Reasoning (LPNMR'01)*, Vienna, Austria, 2001, pp. 434–438.
- [60] TAL, Special issue "Approches sémantiques", *Traitement Automatique des Langues* 35 (1) (1994).
- [61] M. Thielscher, Ramification and causality, *Artificial Intelligence* 89 (1–2) (1997) 317–364.
- [62] G.H. Von Wright, *Norm and Action: A Logical Enquiry*, Routledge & Kegan Paul, 1963, also available at <http://www.giffordlectures.org/Browse.asp?PubID=TPNORM>.
- [63] J. Wu, B.G. Heydecker, Natural language understanding in road accident data analysis, *Advances in Engineering Software* 29 (7–9) (1998) 599–610.