Towards an Information-Based Procedural Grammar for Natural Language Understanding

Alexandre Sévigny
McMaster University

This article describes a number of phenomena which must be taken into account in order to model aspects of information transmission/reception which occur during natural language processing. It presents a brief comparison with two prominent and similar approaches (Dynamic Syntax and Left-Associative Grammar) followed by a detailed description of the model proposed — referred to as Discourse Information Grammar or DIG. It further provides an illustration of the model and a brief discussion of its potential applications to computer-assisted language learning, and concludes with a definition of what is understood by "information" in Discourse Information Grammar.

Cet article décrit un certain nombre de phénomènes que l'on doit considérer afin de pouvoir construire un modèle de la transmission/réception d'information qui a lieu durant tout échange langagier. Il présente une comparaison rapide avec d'autres systèmes de grammaire linéaire semblables, une description assez détaillée du modèle proposé — appelé « Discourse Information Grammar » ou DIG. Il donne ensuite une illustration de DIG, quelques remarques sur des applications possibles du modèle à l'enseignement assisté par ordinateur et le traitement automatique du langage, et finit par une définition du terme 'information' tel que ce concept est utilisé en DIG.

Introduction

Discourse Information Grammar (henceforth DIG) is an approach which sees natural language processing as a linear, nonmonotonic, defeasible process which assembles words into informational units which, in turn, are assembled into yet larger informational units. (*Nonmonotonic* means the result of an operation can be changed and *defeasible* means an operation can be undone.) These assembly processes are subject to various constraints and checks. In essence, DIG represents an effort to begin to simulate how information is assembled and transmitted during natural language exchanges. Fig. 1 shows how the activities and processes utilized in DIG can be represented schematically.

First, it is important to note that the "planes" representing different levels of information accumulated by DIG are not separate components requiring actual movement from one level to another. Rather, the intention is to point

Address for correspondence: Alexandre Sévigny, Communication Studies Programme, Room 507, Togo Salmon Hall, McMaster University, 1280 Main Street West, Hamilton, ON, L8S 4L8. E-mail: sevigny@mcmail.cis.mcmaster.ca.

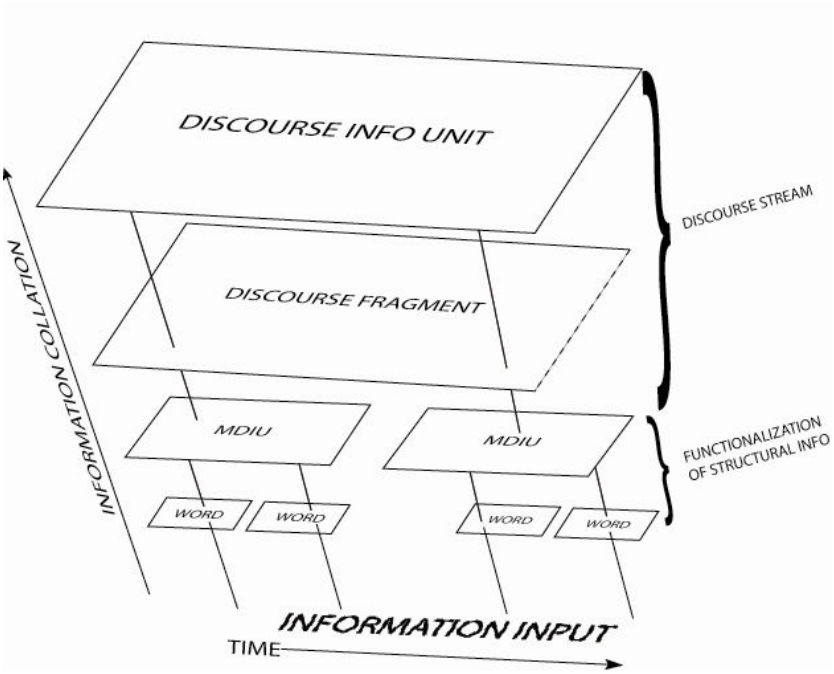

Figure 1: Schematic representation of DIG

out that these "levels" represent temporally differentiated stages of information accumulation. At first input tokens are lexicalized as words. These, in turn, are assembled into structures which, upon being assigned a functional role, become functionalized structures. At this point, they have accumulated enough information to become minimal discourse information units (MDIUs). It is at this stage of "development" or information accumulation that an MDIU can be integrated into the discourse stream. This, as we will see below, triggers new information types. At this point, the words and their context(s) as well as their contraints can enter into the discourse stream as full-fledged members of the stream. If no signal has been processed which would signal the end of input, the newly integrated MDIU forms an incomplete or open-ended discourse stream which is referred to in DIG as a discourse fragment. Eventually, discourse fragments are closed and form a discourse unit, at which point new information is triggered. (See below under A Worked Example for details on how this takes place.)

DIG is lexicalist in orientation and in this way is similar to prevailing notions in current generative linguistic theory. Other examples of lexicalist theories of grammar are Head-Driven Phrase Structure Grammar (Pollard and

Sag, 1994) and Lexical-Functional Grammar (Bresnan, 2002). Lexicalist theories make heavy use of lexical representations in order to constrain potential unit sequences. In order to effect such an approach, it has been necessary to examine closely what is to be understood by the term "information". Information, in the DIG approach, is viewed as an orchestration of several types of information, all of which are closely related to the use of language as the medium of transmission. The following types of information are common: lexical, structural, functional, situational, contextual, logical, semantic, topical, relational and discursive. Each information type contributes values which the other types do not. Combined and networked, these various types of information form what is called the *discourse stream*. Initially, the discourse stream is empty. At this point, it is designated by the symbol $\emptyset$. However, as will be described later, words and units combine and the discourse stream begins to evolve into a network of information states and relations all potentially co-occurring at different stages of specification.

An immediate problem which must be dealt with during linear, incremental assembly of this discourse stream is knowing when one structure ends and the next begins. There are three common cases: final structure, intermediate non-overlapping structures, and nested or embedded structures. The first involves what is called *complete closure*. In written texts, this is usually signalled by one of the four punctuation marks: the period (.), the question mark (?), the exclamation mark (!) and the semi-colon (;).[1] The second involves what is called *partial closure*. This is usually signalled either by structure-type incompatibility and absence of a complete closure marker or by the presence of a partial closure marker. The most common partial closure markers in written texts are the comma (,), the colon (:), ellipsis points (. . .) and the dash (—). The third case involves embedded structures, which are structures that involve structures that are embedded in other structures, often via the use of a subordinating conjunction or some other linking grammatical element. Any model of dynamic, linear grammar requires a mechanism or a series of mechanisms that recognizes when a structure is complete. Such a capability is necessary in order to effect "wrapping up" processes, a critically important step in incremental information accumulation and particularly important in dialogue, where it is necessary to deal with numerous intermediary states or *discourse fragments*.

The idea behind DIG is simple to state but difficult to achieve: words are input one at a time and connected into structures. Structures are assigned functional roles. At this point, the functionalized structure is integrated into the discourse stream. Initially, this is trivial because the initial state integrated with new information always yields some type of information. Unless integration involves complete closure, the discourse stream is incomplete and hence referred to as a discourse fragment. If a new functionalized structure is integrated

into the discourse stream and found to be compatible with its immediate predecessor structure, then the two structures are functionally bound. At this point, discourse information accumulation proper has begun. A bound functionalized structure, integrated fully into the discourse stream becomes an MDIU. Among other values, it adds *situational information*, which is information about participant roles, argument structure and other information necessary to establish context.[2] This process of incremental information accumulation continues until the discourse is closed, at which point we have a *discourse information unit* (henceforth DIU). Often this is equivalent to the sentence, but in many cases — notably dialogue, conversation and interior monologue — fragments are frequent. In the event of a new discourse stream being subsequently initiated, it will be built up structure by structure and eventually combined with the previous DIU. This, very briefly, is the approach symbolized by DIG. Of course, this has been an ideal whirlwind tour of the process. In reality, numerous constraints and checks are used to deal with problems of ambiguity, polyvalence and so forth. A few of these problems will be examined in the section A Worked Example.

The emphasis on information accumulation and collation as the basic activity of natural language processing has necessitated the creation of two particular items:

i) lexical entries designed to work with an information-based approach; and

ii) special units designed to capture information as efficiently as possible, given the real-time constraints involved in actual language processing.

Definitions of these units and entries will be given under The Basic Mechanism and A Worked Example.

There remains only to comment on the scope and applicability of DIG. Briefly, since the DIG approach deals with lexical entries as bundles of information features, it follows that in terms of linguistic theory and general applicability the DIG approach lends itself to any natural language, in spoken or written form. Of course, modifications pertinent to a given language will need to be incorporated as well. Though it seems that languages have a great deal in common, including lexical information and typical use of such information as well as assembly mechanisms and informational content accumulated during natural language processing, it remains to be established whether there will result a substantial common or universal core applicable to most or possibly all natural languages.

The scope of this article is quite restrained, concentrating on a few short examples taken from a restricted corpus of written English and French. Moreover, the work described here does not reflect phonological realism, as exploration into real-time prosodic analysis has indicated that it is possible to set the problem of phonology aside for the time being without compromising the theory underlying the DIG approach (Martin and Sévigny, 2000).

Finally, it is germane to ask whether DIG is a natural language parsing program. In fact, the DIG approach cannot really be called a parser because its purpose is not simply to determine whether a set of tokens can or cannot form a parse tree. There are two basic differences at work:

i) the tokens — that is, words, constructed structures, information units — contain procedural as well as prescriptive information which can affect further analysis — in other words, these tokens are dynamic rather than static; and

ii) the purpose of applying DIG to input is not to generate a parse string or determine whether such a string is licit or not — rather, the purpose is to represent the state of information accumulation.

Whether the text comes to complete closure or not does not affect or nullify the outcome. Such a state of affairs is generally not acceptable in formal parsing. Thus, DIG incorporates a parsing mechanism, but is better described as an information accumulation and collation system. Eventually, as our knowledge of information processing matures, it may well be that more advanced information parsers will be developed, but for the time being that remains a research goal.

Other Models of Linear Grammar and Advantages of DIG

There has recently been a minor explosion of interest in the concept of time-linear grammar in linguistic theory, with several new approaches being developed: Dynamic Syntax (Kempson, Meyer-Viol and Gabbay, 2001), Markov Syntax (Tugwell, 1998), Dynamic Dependency Grammar (Milward, 1994), Linearized Phrase Structure Grammar (Shin, 1987), Left-Associative Grammar (LA-Grammar) (Hausser, 1999) and DIG (Sévigny, 2000, 2002, 2003). To avoid excessive detail, most of which would be beyond the scope of this paper, I will only compare LA-Grammar and Dynamic Syntax to DIG.

DIG resembles both Dynamic Syntax (Kempson *et al*., 2001) and LA-Grammar (Hausser, 1999) in that all three approaches analyse language on a linear, left-to-right basis, which means that words are processed as they are input and representations assembled as soon as possible. All three approaches are formal explorations into what is needed to design a grammar which models natural language processing dynamically. Fundamental to this process is the recognition that natural language understanding operates in time, and that

words are input one at a time. It is this temporal linearity which is referred to by the expression "linear, left-to-right", which can also be read as first input to next input. This places all three approaches being discussed squarely in the performance grammar domain, although it is conceivable, and even likely, that results yielded from an increased awareness and understanding of performance grammar phenomena will contribute to our understanding of competence as well, especially given the novel characterizations of traditional generative linguistic problems such as cross-over, *Wh*-questions, anaphora resolution, quantification (Kempson *et al*., 2001) and periphery phenomena (Cann, Kempson and Otsuka, 2002) being proposed by linguists using Dynamic Syntax.

In spite of sharing a number of common traits, however, DIG, Dynamic Syntax and LA-Grammar differ in approach, method, goals and theoretical stance. The design of LA-Grammar was strongly influenced by its computational linguistics background and its author's focus on mathematical completeness and efficiency. As such, it operates on a finite-state backbone with rule packages, finite state transitions, valencies, and so forth. LA-Grammar is conceptualist and cognitivist in its orientation, as Hausser's aim is to construct a cognitive agent (a robot) which can interact linguistically with humans in a realistic fashion. To achieve this instantiated cognitive linguistics, he has constructed a theory which he calls SLIM (Surface Compositional Linear Internal Matching [Hausser, 1999, p. 8]) which operates using an innovative database metaphor for procedural semantics that entails the creation of an ontology of conceptual type structures in a relational database (Hausser, 1996). These types, in turn, are used to construct a semantic and pragmatic context within which understanding of an utterance can be situated and analysed.

There are several immediate limitations of LA-Grammar from the perspective of incremental information accumulation. First is the fact that there is no real concept of structure inherent in the model. Conceptually, syntax is simply the matching of the first element and the remainder of a list. A second limitation concerns the property of brittleness. Basically, if a new input arrives which is positionally not acceptable, either it is rejected and the utterance is rejected or numerous distributed changes have to be made across various rule packages in order to accommodate this novelty of pattern or type. A third limitation follows immediately from the second: LA-Grammar is form-based rather than information-based. This property will eliminate many fragments as well as slightly incorrect grammatical utterances even though they would normally be processed by a human processor. It is not immediately clear how this shortcoming would be dealt with in current LA-Grammar (Gelbukh, 1999). This does not mean that LA-Grammar could not be applied to the solution of traditional linguistic problems, but that has not been Hausser's interest or orientation. This relatively functionalist orientation is in contrast with the approach taken in Dynamic Syntax.

Dynamic Syntax comes from the domain of formal logical analyses of language, from the collaboration of Ruth Kempson, Dov Gabbay and Wilfried Meyer-Viol, in an effort to incorporate logical analysis and the proof-theoretic understanding process described in Chapter 2 of Sperber and Wilson (1995). As such, it is unlike its model-theoretic counterparts in that it does not adhere to the notion that natural languages are formal languages. Though it transcends these approaches in range and simplicity of structure and rules, Dynamic Syntax still operates in terms of formal structures derived from logic and mathematics, using *labelled deductive systems* (Gabbay, 1996) and the *logic of finite trees* (Blackburn and Meyer-Viol, 1994) as the grounding for its formalism. For instance, it uses a decorated, binary (partial) tree structure as the fundamental construct to attain interpretations. Essentially, an initial state triggers a goal to attain a unique interpretation. This interpretation is gradually attained by building up a labelled binary tree node by node, with each new word triggering either a new node, an update on a node or a new formula.[3] This process of enriching a node or adding a new one follows formal requirements which are not necessarily based on informational content since "DS does not model real time parsing, but characterizes the body of knowledge required for incrementally building interpretations" (Marten, 2002, p. 35, n. 11). The resulting decorated tree, a history as well as an accurate representation of the logical form of the utterance just processed, is called an interpretation of the utterance. In order to build these complete trees, Dynamic Syntax also uses position pointers which can move up or down the nodes of the tree in progress. This flexibility allows Dynamic Syntax to deal with many functional and relational phenomena without having to resort to the movement metaphors often used in the minimalist program (Chomsky, 1995). In addition to pointers, Dynamic Syntax also uses "IF THEN ELSE" rules with simple operators to represent lexical entries. Formulae, rather than words or names, are used to label tree nodes. These formulae may be temporarily underspecified until the necessary missing information is processed or becomes derivable. Once input stops, any underspecified formulae are filled and, if necessary, moved up to decorate their (immediate) parent node. Because it is oriented toward generating formal interpretations of input, Dynamic Syntax is free to generate pathways which are motivated by tree generation and decoration rather than by information accumulation. From this perspective, resolving formulae down to a primitive type in a proof-theoretic manner can take precedence over the modelling of information accumulation. Because of this possibility, Dynamic Syntax may not always be optimally suited to modelling information accumulation and immediate recording of such information in network representations.

Unlike both LA-Grammar and Dynamic Syntax, DIG is information-based rather than form-based or procedure-based. Moreover, DIG uses two information-based concepts which are absent from the other two approaches:

i) the notion that lexical entries contain sufficient default information to generate and/or enter into locally constrained networks of information; and

ii) the notion that there are different types of units, each endowed with information categories which are additional and complementary to other units.

When information reaches the point where one of these units can be synthesized, the effect is reminiscent of triggering or "firing" in neural networks. Not only is the information synthesized at this new level, but new information categories are brought into active existence. This approach requires a nonmonotonic approach, since earlier information may need to be updated or changed in various ways. Moreover, DIG cannot be said to be a parser in the traditional sense. Nor can it be it be regarded as a strictly formal, proof-theoretic approach since it is concerned with presenting the status quo of whatever information has accumulated up to a given point. If this information is incomplete following some interruption the utterance will not be rejected. Thus, DIG can handle a wide variety of discourse types, including conversation. If the informational content becomes problematic it is usually because it has lost coherence to the point of becoming meaningless or contradictory even to human processors.

The concept that intermediate states or stages are important in grammatical derivation is not unique to DIG. LA-Grammar operates mostly with intermediate states until an utterance is complete or is dismissed as incorrect. Dynamic Syntax uses intermediate states extensively also, as mentioned earlier, in the form of partial subtrees whose nodes are not labelled with words, but with incomplete decorations consisting of underspecified functional formulae. The latter are similar to the functional-role assignment operations employed in DIG, but differ in that Dynamic Syntax is designed to attain the goal of decorated binary tree representations, while DIG does not require this property since it accumulates and carries its baggage of information incrementally and builds networks of information.

DIG generates very clear, concrete lexical requirements and in doing so is contributing to a deeper understanding of exactly what data structures and feature complexes are needed in a lexicon designed for formal modelling of natural language processing. The unification of attribute feature values must operate with something like fuzzy values and some probability mechanism. Thus, the features can differ within certain limits and still be compatible. Just exactly what these limits are and what the ranges are remains an intriguing area for further research. Moreover, by accumulating this information, DIG provides a fairly detailed definition of what is to be understood by its central term: "information".

Basic units and processes used in DIG

The generic lexical entry for a word type will be referred to as a lexeme and defined in terms of a template of *feature structures* with at least five fundamental attribute-value sorts. An example of a partial lexical template is given in (1).[4]

(1) NAME: ⟨ ⟩
CATEGORY:
INDEX: gender [ ], number [ ], person [ ]
STRUCTURE-TYPE: [ ]
SEM: {. . . }

NAME simply refers to a printed word. Each lexical entry belongs to a CATEGORY, such as noun, verb, linker, pronoun. In the case of homophones, each will have a different category; in the case of polysemy, an entry will have a range of entries, each with differing default information. Each category belongs to one or more STRUCTURE-TYPES, such as nominal structure and verb structure.[5] INDEX refers to features such as gender, number and person. Some languages would use other features, such as animate and inanimate, when they are syntactically relevant. The SEM {. . . } field is an open set of descriptive features, usually binary-valued, although there appear to be ternary-valued features as well. The ellipsis points signify that the SEM field is an editable open list, which is a necessary condition to accommodate dynamic incremental information accumulation and may be altered to permit adaptation to immediate contextual requirements and/or constraints.

How DIG moves from token to specified word

When a printed word is first input, it is merely a *token*. Once it has been located in the lexicon, it receives minimal lexical information and becomes a *word*. As mentioned previously, the term *lexeme* is used to refer to entries in the lexicon. Since it is common to edit during natural language processing, DIG uses four levels of specification: default, underspecification, partial specification and complete specification. Default specification usually corresponds to hard-wired grammatical information, such as occurs in the INDEX values for the French definite article *le* ('the') which has a default specification of [+singular] and [+masculine]. As information accumulates, various participating units will receive additional specifications, some of which may entail corrections of previous specifications. This is necessary in order to capture information accumulation in a time-linear fashion.

Basic Structure Types

A structure is defined as a quintuple:

(2) ⟨HEAD, F-SET, S-TYPE, STATE, TEMP⟩

HEAD is the most important element in the structure. It collects all the attribute-features which accumulate during the construction of the structure. The intent here is to simulate a person's ability to replace entire structures with single, say, anaphoric references. In some cases, the HEAD sets feature constraints which must be met by other structures which become linked with it in some way. For instance, the verb structure "eat(ns1, {ns2}, {compl})" requires that ns1, the first nominal structure, be normally marked [+animate], and that ns2, the second nominal structure, be [+foodstuff], etc.

An F-SET is a small range of possible functional roles that a structure may potentially assume. The idea is to mimic the human ability to analyse a HEAD and related words as a unit. Once a structure has been completed, it is closed and can be used in many different ways, especially if it is a nominal structure. One of the reasons for using one language as a metalanguage is to accentuate the probable reality that humans share a (large) common core of ideational concerns. It is through studying this core explicitly in various languages that we may hope to attain a universal understanding of natural language understanding/performance and potentially of language competence.

Each lexical entry has a pre-specified default structure type value (its S-TYPE), learned during language acquisition and stored in long-term memory. This feature allows for modelling the intuitive notion of immediate recognition that words do or do not belong together as type units. In DIG, the S-TYPE feature plays a critical role, given the need to establish unit boundaries and scope. Examples of S-TYPE are nominal structure, verbal structure and adjectival structure.

When a structure is triggered, its STATE feature is marked "open". When the structure comes to an end, STATE is marked "closed". Only then can the structure be assigned a contextually constrained F-ROLE value from its F-SET. Note that we often can anticipate what the F-ROLE value will be as soon as we know the structure type currently being developed. For example, I know that *the* has initiated a direct object in the partial utterance "*I was watching the . . .* " because in this case the article *the* initiated a nominal structure. However, I still need to know what the HEAD of the nominal structure will be.

Initially, TEMP is an empty set of attribute features. As a structure is built up, accumulated attribute features are copied into TEMP. Note that the NAME attribute of the words contained in the structure is not accumulated. When the HEAD of the structure has been processed, TEMP is unified with HEAD. Afterwards, additional attribute-feature values are unified with the HEAD directly. Once this is done, TEMP is de-activated. The intent here is to mimic the human ability to store attributes even before it is known what to connect these attributes to. For example, in the sentence *It was a smelly, warped, twice-broken, over-used, rat-infested, old . . . CABIN*, once the first adjective is parsed, we know a noun head will eventually follow.

Discourse Units

In addition to structure types, there several discourse-information types, that is, types of informational units that are connected to the discourse stream of the text. Four of these types occur regularly: *functionalized structure*, *minimal discourse information unit* (MDIU), dependent *discourse fragment* and independent/complete *discourse information unit* (DIU).

There is an information continuum being accumulated in these units which range from functionalized structure to MDIU to discourse stream to discourse fragment to DIU. For instance, before a simple structure can be integrated to a discourse stream, the former must be *functionalized*; that is, it must be assigned one of the values from its F-SET. Once functionalized, the simple structure specifies the start of relations and becomes an MDIU enriched with a few additional parameters which reflect that it is ready to become an integral part of the discourse stream and more than a mere disembodied structure. Before the structure can become fully realized as an integral part of the discourse stream, it must undergo F-ROLE value unification. While it lacks F-ROLE value unification, it anticipates rather than describes a situation where one or more participants will become more fully specified. Implied is the notion that discourse contains more than mere structural and functional information. In concrete terms, the nominal structure *the little white cat* is merely a name, not discourse. Later, when more information is processed, we learn it is the subject and discourse has begun. Intrinsic to discourse is such information as participant role(s) and relation(s). Although this may remind one of case theory at the sentence level, the discourse stream is more flexible and may terminate without reaching full sentence status. In such cases, the discourse ends as a discourse fragment and its accumulated information load is stored. Though unusual in written expository prose, discourse fragments are a common phenomenon in written dialogue and monologue.

It should also be noted that the information is cumulative in that it represents not a flushing and moving of structures, but additional layers of information. Consequently, it is also possible to view these categories in terms of subsumption, since we can order the unit types, in terms of their information content, as follows:

(3) word $\subseteq$ structure $\subseteq$ f-structure $\subseteq$ MDIU $\subseteq$ d-frag $\subseteq$ DIU

Functional Role Sets

Listed below for convenience is a brief summary of the F-SET for the nominal structure and the verbal structure. Beside each structure type is a list of the common functional roles it may assume. Not all languages use all structure types. Nor do they always assign functional roles in the same manner. Nevertheless, the concept of structure type and functional role(s) is common to all

languages and seems to be a constraint on how human information representation operates.

(4) *nominal structure*: topic, subject, direct object, indirect object, object of pre-/postposition, subjective completion, apposition, comment
verbal structure: predicate, auxiliary, pro-verb

The functional roles have been extrapolated from a combination of generative grammar, traditional grammars and pragmatics. They have been known for a long time and do not constitute new theoretical information. What is new is the application of such information in a formal context for the construction of a left-right discourse information grammar. These functional roles are at the heart of our capacity to represent situational/pragmatic information, an ability which is essential to capturing the flow of information in discourse as it accumulates during discourse processing. Functional roles constitute both declarative and procedural information, subject to constraints imposed by the Adjacency Constraint Principle described below.

Functional-role assignments also permit the creation of anticipated templates which in turn help to effect structure selection. The DIG approach makes heavy use of anticipation phenomena. This is necessary in order to deal with time-linear constraints. If too much active, dynamic information accumulates, processing will be compromised. The approach behind DIG assumes that a lot of decision-making is speeded up because the information processed has been reduced in various ways. This is an important property because it reduces the time needed to process new information. The intention here is to mimic the human ability to know almost immediately whether a newly input word fits in with the current structure and/or whether the newly initiated structure type can immediately follow the current structure type now being closed. For example, the fragment, *the cat with my* is possible while *the cat with my was* is immediately rejected as soon as *was* follows the determiner since the preposition *with* will normally be followed by a nominal-structure element.

There are two points which bear repeating concerning F-ROLE value sets attached to structure types. First, an F-ROLE value cannot be assigned to a structure until the structure is complete, that is, until it is closed. The reason for this need for structural closure is that in linear, incremental left-to-right assembly, there is no top node from which to monitor progress. Such information must be signalled via input itself in some manner (see A Worked Example below for a discussion and illustration). Second, a structure is closed by type-incompatible input. For example, when a nominal structure is initiated, it cannot receive a finite verb, as in (5).

(5) [$_{ns}$ the little cat [$_{vs}$ watched . . .

We are now faced with two problems:

i) we have no way of knowing in advance what kind of construction is being assembled; and

ii) we do not want to lose any information accumulated to date.

There must be a way of accomplishing three sequential processes:

i) input of incompatible input will initiate a new structure type;

ii) the old structure must be closed; and

iii) the old structure must receive its F-ROLE value as soon as possible.

The solution lies in using a small set of operators repeatedly. These operators locate arguments in the lexicon, assign functional roles and indicate when structure is opened, built up or closed, for example. A few examples and brief descriptions of basic operators are given in the Appendix. See A Worked Example for details on how they are applied.

The Adjacency Constraint Principle (ACP)

Fundamental to linear discourse assembly and working closely with the basic operators is the *Adjacency Constraint Principle* (ACP) which can be stated briefly as follows:

A word constrains the type of word which may follow it in three ways:

i) paradigmatically (its word type);

ii) syntagmatically (its structural type value); and

iii) semantically (feature compatibility).

Given the ACP, it follows that a new word introduced into an utterance stream triggers one of the following conditions:

i) type-compatibility with the structure type currently being built;

ii) type-incompatibility with the current structure type but occurring at a point when the current structure type can be closed or paused. A structure can be closed if it possesses or can be assigned a HEAD member, using the operator *change_to_name_type* (arg), if it is possible to do so.

iii) being an embedding or coordinating linker, in which case embedding or s-pausing will occur;

iv) type incompatibility with no possibility of closure. This generates an error.

The Basic Mechanisms

The approach used in DIG views discourse as consisting of words which enter into primary functional relationships, as opposed to words merely being cited

or named. Functional relationships occur at different discourse levels; primary functional relationships occur at the *default level*, or *non-embedded level*. The term *default level* or non-embedded level refers to non-embedded words and expressions. For instance, the structure in (6) contains several functional relationships but the structure, with the embedded structure *que tu as mentionnés* ('which you mentioned') removed, reduces to *les éléments* ('the elements'):

(6) les éléments que tu as mentionnés
the element-PL that you-2nd.SG have-2nd.SG mention-PPT
'the elements that you mentioned'

At this instance in communication, *les éléments* ('the elements') is merely a name and does not constitute discourse since it could be a topic, a subject, a direct object, a comment, etc. For discourse to begin, names must be assigned a primary or non-embedded functional relationship, which may or may not involve a predicate. When one of these functional relationships is assigned to *les éléments* (*que tu as mentionnés*) ('the elements (which you mentioned)'), then discourse has begun, as in (7).

(7) Les éléments que tu as mentionnés sont intéressants
the element-PL that you-2nd.SG have-2nd.SG mention-PPT are-3rd.PL interesting-PL
'The elements that you mentioned are interesting.'

where *sont intéressants* ('are interesting') allows the string *Les éléments que tu as mentionnés* ('The elements that you mentioned') to be assigned the functional role of subject.

A Worked Example

It is now time to put all this apparatus to work. For ease of exposition, only relevant features will be listed. The sequence to be analysed is presented in (8).

(8) Le petit chat blanc a mangé la souris. Il avait faim.
The-MASC.SG little-MASC.SG cat-MASC.SG white-MASC.SG have.3rd.SG eat-PPT the-FEM.SG mouse-FEM.SG. He-3rd.MASC.SG have-IMP.3rd.SG hunger.
'The little white cat ate the mouse. He was hungry.'

In order to simulate the nature of linear, incremental input as well as the effect of information accumulation, input will "arrive" one token at a time. Thus, our first example begins with token input *Le* ('the'). This token yields two possibilities from the lexicon, which are shown in (9) and (10).[6]

(9) NAME: ⟨**Le**⟩
CATEGORY: definite article
INDEX: gen [**+masc**], num[**+sg**], pers[**3rd**]
STRUCTURE-TYPE: **nominal structure**
STRUCTURE-TYPE-HEAD? **no**

F-ROLE: specify(HEAD): [indicates that the HEAD will be marked for definiteness]
SEM: {DEFINITENESS: [**+definite**], . . . }

(10) NAME: ⟨**Le**⟩
CATEGORY: personal pronoun
INDEX: gen [**+masc**], num[**+sg**], pers[**3rd**]
STRUCTURE-TYPE: **nominal structure**
STRUCTURE-TYPE-HEAD? **yes**
F-ROLE: direct object: (HEAD)
SEM: {variable}
ANAPHORIC LINK: **Le = ??**

Some of these values are default values; for example, the assumption that the INDEX: *person* value is 3rd. All these choices may be overridden if necessary. To simplify references, this nominal structure will be identified as `ns1`. Applying the operator "[" to *le*, as in (11), opens `ns1`, and yields the results shown in (12) or (13), where "??" represents possible input:

(11) [(Le)

(12) [Le + ??⟨mod, n⟩ . . .
`ns1`

In this case, *le* is being treated as a definite article initiating a nominal structure.

(13) [Le + ??⟨$\texttt{vs}_{\text{+finite verb}}$⟩ . . .
`ns1`

In example (13), *le* is being treated as a personal pronoun used as a pre-posed direct object.

In both cases, (12) and (13), the structure remains open, symbolized by the suspension points, because we have no information as yet to effect closure. This situation models the uncertainty a receiver would feel if the word *le* were followed by a prolonged hesitation, as though the sender were pondering or hesitating. Notice that the concatenation operator "+" is activated. It will receive the new input. In terms of TEMP, `ns1` has now accumulated one of two representations, either (14) or (15):

(14) TEMP-1:
CATEGORY: definite article
INDEX: gen [**+masc**], num[**+sg**], pers[**3rd**]
STRUCTURE-TYPE: **nominal structure**
STRUCTURE-TYPE-HEAD? **no**
F-ROLE: det(HEAD): (HEAD): [+definite]
SEM: {DEFINITENESS: [+definite], . . . }

(15) TEMP-2:
CATEGORY: personal pronoun
INDEX: gen [**+masc**], num[**+sg**], pers[**3rd**]
STRUCTURE-TYPE: type: **nominal structure**
STRUCTURE-TYPE-HEAD? **yes**
F-ROLE: direct object: (HEAD)
SEM: {variable}

Next, we input *petit* ('little') which yields (16).[7]

(16) NAME: ⟨**petit**⟩
CATEGORY: descriptive adjective
INDEX: gen [**+masc**], num[**+sg**], pers[ ]
STRUCTURE-TYPE: type: **nominal structure**
STRUCTURE-TYPE-HEAD? **no?** [Normally **no**, but may be nominalized as the HEAD]
F-ROLE: **modification:adjectival:description**(??HEAD)
SEM: {**SIZE**: <**normal**$_{\text{HEAD}}$}

Because the object pronoun form is not followed by a modifier, this new input eliminates *le* as a personal pronoun and that path is abandoned. TEMP is then analysed as in (17).

(17) INDEX: gen [+masc], num[+sg], pers[3rd]
STRUCTURE-TYPE: type: nominal structure
STRUCTURE-TYPE-HEAD? no? ['?' because 'petit' could be nominalized]
F-ROLE: mod(HEAD)
SEM: {DEFINITENESS: [+definite], **SIZE: <normal**$_{\text{HEAD, }\ldots}$}

It is interesting to note that each word actually contributes a fairly small amount of information which often carries a lot of redundancy. This is meant to simulate and partly explain our ability to process new input very quickly. It seems as though language as an information-transmitting system uses redundancy regularly. As the example builds up, we will see more instances of informational redundancy. We now input *chat* ('cat') which yields (18).

(18) NAME: ⟨**chat**⟩
CATEGORY: noun
INDEX: gen [**+masc**], num[**+sg**], pers[**3**]
STRUCTURE-TYPE: type: **nominal structure**
STRUCTURE-TYPE-HEAD? **yes**
F-ROLE: ?
SEM: {**[+animal], [+feline], [+domestic], [+animate],** . . . }

Since *chat* is type compatible with nominal structure, which is the current structure type being assembled in the structural buffer, and since *chat* can assume the HEAD value, TEMP will unify destructively with *chat* to yield (19).

(19) NAME: ⟨chat⟩
CATEGORY: noun
INDEX: gen [+masc], num[+sg], pers[3rd]
STRUCTURE-TYPE: type: nominal structure
STRUCTURE-TYPE-HEAD? **yes**: HEAD = 'chat'
F-ROLE: ?
SEM: {DEFINITENESS: +definite, SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate], . . . }

At this point, TEMP is now empty. Continuing, token input *blanc* ('white') yields (20), which unifies directly with *chat* to yield (21):

(20) NAME: ⟨**blanc**⟩
CATEGORY: descriptive adjective
INDEX: gen [**+masc**], num[**+sg**], pers[ ]
STRUCTURE-TYPE: type: **nominal structure**
STRUCTURE-TYPE-HEAD? **no** [because in post-HEAD position]
F-ROLE: mod(HEAD)
SEM: {**COLOUR: [blanc]** . . . }

(21) NAME: ⟨chat⟩
CATEGORY: noun
INDEX: gen [+masc], num[+sg], pers[3rd]
STRUCTURE-TYPE: type: nominal structure
STRUCTURE-TYPE-HEAD? **yes**: HEAD = 'chat'
F-ROLE: ?
SEM: {DEFINITENESS: [+definite], SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate], **COLOUR: [blanc]** . . . }

If we now input *a* ('has'), we trigger a flurry of activities, most of which occur repeatedly either in parallel or sequentially at such a rate that we are largely unaware of them, thanks to the repetitiveness of their contents and nature as well as the thousands upon thousands of hours we have practised them. To go back to basics, just observe what you experience when you make your first attempts at spontaneous conversation or reading in a second-year language course, particularly in a language which differs from yours in significant ways. Eventually, of course, you will attain a smooth flow of information processing but this fluency requires a considerable investment in time and practice before becoming established. What follows now is a quick description of a brief instance of this type of flurry of activities.

First, the input *a* ('has') is lexicalized and receives the value "verb structure" as the value for its S-TYPE field. This makes it incompatible with the nominal structure currently being assembled. Secondly, *a* is recognized as being of a type which does not signal embedding. Following (or possibly accompanying) this new information, *a* now triggers the operator "[( )" with argument *a* and yields a new current structure: [$_{\texttt{vs1}}$ **a** + ?? . . . . This third step triggers closure of ns1, indicated in DIG as "](ns1)". A fifth activity occurs when "](ns1)" triggers check-adjacency condition(ns1, vs1). The immediate adjacency between ns1 and vs1 signals the F-ROLE value subject for ns1. However, there is as yet no unification occurring because functional-role assignment is not yet possible. This simulates our "inner searching" for the functional role of this particular nominal structure. Following a bit of cogitation, an almost immediate result of effecting the operator FRA(ns1) yields the value subject for ns1. This is represented as the functionalized structure ns1$_{\text{subject}}$, which now becomes MDIU-1. Among other things, attainment of this structural status triggers the specification that ATTENTIONAL FOCUS is *chat* ('cat'). The establishment of MDIU-1 also triggers the process of integration into the discourse stream: $\oplus$ ($\emptyset$, MDIU-1), where $\emptyset$ designates the default starting point. When MDIU-1 integrates with the default discourse stream $\emptyset$, the newly started discourse stream will be assigned the status of discourse-fragment-1 since MDIU-1 was not followed by a terminator. This simulates our ability to know that the incoming information is not yet complete. Note also that MDIU-1 has not yet unified with anything in the discourse stream since it has nothing with which to unify at this point.

Integration of MDIU-1 into the discourse stream as discourse-fragment-1, however, brings in additional parameters of information: ⟨situation, intention, logical structure, semantic field, topic chain⟩. At this stage, the information is scant: we know that *le petit chat blanc* ('the little white cat') seems slated to be the doer of some upcoming event. What that event is we have still to learn, along with any additional bits of information which may be attached to it. Right now, if the speaker were to stop, we would be left with a lot of questions. This state of affairs can be schematically represented as follows (22 and 23):

(22) SITUATION: {ns1 **= participant1; F-ROLE value: subject, participant2 ??** . . . **, relation(s): ??** . . . **time: ??** . . . }
INTENTION: **ASSERTION**
LOGICAL STRUCTURE: **P(x** . . . **)**
SEM: {. . . }

From the MDIU, we obtain more information, which yields (23).

(23) 'chat': **SIZE: <normal**$_{\text{HEAD}}$**, [+animal], [+feline], [+domestic], [+animate], COLOUR: [blanc]** . . . }
TOPIC CHAIN: **chat**$_{\text{subject}}$ **— ??**

Combining these two states of information yields the fuller accumulation, illustrated in (24).

(24) SITUATION: {ns1 = participant1; F-ROLE value: subject, participant2 ?? . . . ,
relation(s): ?? . . . time: ?? . . . }
INTENTION: ASSERTION
LOGICAL STRUCTURE: P(x . . .)
SEM: {SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate],
COLOUR: [blanc] . . . }
TOPIC CHAIN: chat$_{subject}$ — ??

If we continue with *mangé* ('eaten') we resolve the ambiguity of whether *a* was an auxiliary or main verb since it is clearly [+aux] in this case. Also, *a* is marked [+sg, +3rd], features which must be matched by ns1 in "manger(ns1, {ns2, compl})". So far, this new structure, vs1, has yielded the information presented in (25), which unifies with (26) to yield the results shown in (27).

(25) NAME: ⟨**a**⟩
CATEGORY: verb
INDEX: num[**+sg**], pers[**+3rd**]
STRUCTURE-TYPE: **verb structure**
STRUCTURE-TYPE-HEAD? If *a* = vfin then **yes**; if *a* = aux then **no**
F-ROLE: ?
SEM: {. . . }

(26) NAME: **mangé**
CATEGORY: verb
INDEX: num[**+sg**], pers[+3rd]
STRUCTURE-TYPE: **verb structure**
STRUCTURE-TYPE-HEAD? If *mangé* = vfin then **yes**; if *mangé* = participle then **no**
F-ROLE: If *mangé* = vfin then **predicate**; else **modifier**
SEM: {manger(ns1, {ns2}, {compl})}

(27) NAME: **a** + **mangé**
CATEGORY: verb
INDEX: num[**+sg**], pers[**+3rd**]
STRUCTURE-TYPE: **verb structure**
STRUCTURE-TYPE-HEAD? mangé;
F-ROLE: ?
SEM: {manger(ns1: [+animate], . . . , {ns2, {compl}})}
TEMPORAL: [+past], [+punctual], [+complete]
EVENT-TYPE: [+process]

Note that we still cannot integrate this second structure (27) into the discourse stream because we still do not know for certain whether it is complete or not.

If we now continue with input *la* ('the'), this latter will be processed as ns2. This will cause vs1 to be closed, assigned the functional role 'predicate', made into MDIU-2 and integrated into the discourse stream. Since it can bind with ns1 the latter assumes the functional role of subject and unifies with variable ns1 in 'manger(ns1: [+animate], . . . , {ns2, compl})'. Furthermore, vs1 can be integrated into the discourse stream which now becomes discourse-fragment-2, since a terminator was not processed. So far, this discourse stream can be schematized as in (28).

(28) D = {discourse-fragment1 ⊕ discourse-fragment2 ⊕ ??}

To recapitulate: in this case, the process of integration has added one additional piece of information: the unification of ns1 with vs1 which in this case effects the assignment of the F-ROLE value $ns_{subject}$ to ns1 for vs1 (manger(ns1= $chat_{subject}$, {ns2, {compl}})). Since the two nominal structures are feature-compatible as far as the minimal indicated requirements are concerned, the assignment is successful. Had there been a problem, the process would have stopped here and a correction made or demanded. The information accumulated so far is schematized below as (29). Again, bolded information represents new information.

(29) SITUATION: ns1 = participant1; F-ROLE value: subject
participant2 ??. . .
relation(s): **predicate**: **manger(participant1 = ns1, x)**
time: **past, punctual, complete**
INTENTION: ASSERTION
LOGICAL STRUCTURE: P(x . . .)
SEM: {'chat'
SIZE: <$normal_{HEAD}$, [+animal], [+feline], [+domestic], [+animate],
COLOUR: [blanc]}
TOPIC CHAIN: $chat_{subject}$ — **manger(**$\mathbf{chat_{subject}}$**, ?? . . .)**

Note that we have processed information pertaining to the relation 'predicate', including a bit of information concerning the time and type of this particular event. Additionally, we have now received information pertaining to the colour of the cat. If we now complete this short discourse with *souris.* ('mouse.'), the input *souris* continues ns2 *la* + *souris*. When the terminator "." is processed, however, several processes are triggered:

i) ns2 is closed, assigned the functional role of direct object, made into MDIU-3, integrated into the discourse stream and because it is feature-compatible with ns2 in 'manger(ns1: [+animate], . . . , ns2, {compl}), it

unifies with it. At this point, it fully integrates as discourse-fragment-3. In terms of the rest of the information, we now have the information presented in (30).

(30) SITUATION: ns1 = participant1; F-ROLE value: subject
participant2 = ns2: **F-ROLE value: direct object**
relation(s): predicate: manger(participant1, **participant2**)
time: past, punctual, complete
INTENTION: STATEMENT
LOGICAL STRUCTURE: P(x)
SEM: {'chat': **manger(chat, souris)**
SIZE: <normal$_{\text{HEAD}}$, [+animal], [+feline], [+domestic], [+animate],
COLOUR: [blanc] . . . }
TOPIC CHAIN: chat$_{\text{subject}}$ — manger(chat$_{\text{subject}}$, **souris**$_{\text{direct object}}$)

ii) the current discourse-fragment chain (*le + petit + chat + blanc*) ⊕ (*a + mangé*) ⊕ (*la + souris*) becomes DIU-1. This adds the additional information parameter "discourse-type", which is realized as NARRATION/DESCRIPTION for the following reasons:

a) the intention type was ASSERTION;
b) the logical structure closed as P(x) where x = DIU-1;
c) the rhetorical relation reduced to a simple statement, since there were no indications of any other relationship(s), logical or otherwise.

The end result of the processing is a final discourse information unit that represents an information state which fulfills all constraints and is thus considered well-formed. A little discourse information has already begun to be accumulated, which could serve to constrain possible acceptable input in subsequent representations.

For the sake of illustration, a second discourse unit will now be processed and integrated with the first unit. This will allow us to see how the phenomena we are describing operate across discourse units. Continuing, then, token input *Il* ('He') yields the analysis presented in (31).

(31) NAME: ⟨**Il**⟩
CATEGORY: personal pronoun
INDEX: gen[**+masc**], num[**+sg**], pers[**3rd**]
STRUCTURE-TYPE: **nominal structure**
STRUCTURE-TYPE-HEAD? **yes**
F-ROLE: subject(HEAD)
SEM: {**?** . . . }

Being of the category personal pronoun, *Il* triggers a search for an antecedent which must match it in terms of INDEX values. By default, it will normally establish an anaphoric relation with the nearest compatible predecessor. Once such a candidate is found, unification occurs and *il* will assume the attribute-feature values of its antecedent. It is for this reason that the SEM field above is marked {**?** . . . }. Once an appropriate antecedent is found, the SEM field will be updated. This is meant to simulate the brief hesitation which is felt upon hearing *il* after a pause. The reader/listener is searching for something to attach to *il*. Note also, that in this case, the functional role set is narrowly constrained by the shape of the input, because it is in the nominative case. It is essentially limited to the subject role.

Because *il* follows immediately after the terminator "." of DIU-1, DIU-2 will be linked to DIU-1 by the *null* linker. By default, the null linker normally signals continuation or elaboration. Because of the null link establishing a connection between DIU-1 and DIU-2, the anaphoric link required by the personal pronoun *il* will be established as soon as possible. By default, the antecedent will be the nearest compatible predecessor. Should this specification turn out to be infelicitous, it can be altered. The intent here is to model our need to effect specifications as soon as possible, combined with our ability to change our understanding on the fly concerning what is happening. In this case, the search will begin in the predecessor DIU-1. Since the nearest compatible antecedent having the same index values [GENDER: masculine] is *chat*, *il* binds with *chat* yielding (32).

(32) NAME: ⟨**Il**⟩
CATEGORY: personal pronoun
INDEX: gen[**+masc**], num[**+sg**], pers[**3rd**]
STRUCTURE-TYPE: **nominal structure**
STRUCTURE-TYPE-HEAD? **yes**
F-ROLE: specify(HEAD): [Indicates that the HEAD will be marked for definiteness]
ANAPHORIC LINK: **Il = chat**
SEM: {**DEFINITENESS: [+definite], SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate], COLOUR: [blanc]** . . . }

Since, the anaphoric link connects *Il* with *le petit chat blanc* of DIU-1, the situation has not changed. Subsequent information will therefore be accumulated to the situational information accumulated during the processing of DIU-1. This situation blending is meant to simulate the sense of simplicity we experience during straightforward information transmission, yet another example of what could be called communication economy. Continuing with *avait* ('had-IMP') yields (33).

(33) NAME: ⟨**avait**⟩
CATEGORY: verb
INDEX: num[**+sg**], pers[**+3rd**]
STRUCTURE-TYPE: **verb structure**
STRUCTURE-TYPE-HEAD? **avait**
F-ROLE: ?
SEM: {**avoir**(ns3**: [+animate],** . . . **,** {ns2**,** {**compl**}})}
TEMPORAL: **[+past], [+durative]**
EVENT-TYPE: **stative:description**

The processing of the input *avait* generates a series of processes which, in essence, are almost identical to the flurry of activities we saw above when input *a* ('has') was processed. What is being captured in this aspect of the simulation effected through the mechanisms used in DIG is the fact that normal linguistic communication involves a tremendous amount of repetition. This is undoubtedly necessary if we are to keep up in real time with the flow of information which normally accompanies linguistic activity. Since, the activities are so similar, they will be described here in point-form only:

i) *avait* is recognized as being type-incompatible with ns3
ii) *avait* is recognized as being of type verbal structure and does not signal embedding
iii) *avait*: STRUCTURE-TYPE triggers [(avait) and yields the new current structure $[_{vs2}$ avait + ??
iv) *avait* therefore triggers closure of ns3:](ns3)
v)](ns3) triggers check-adjacency-condition(ns3, vs2)
vi) *check-adjacency-condition(*ns3*,* vs2*)* shows that in this case, lack of any pause between ns3: vs2 triggers the F-ROLE value of subject for ns3. There is no unification occurring yet.
vii) FRA(ns3) yields the value **subject** for ns3, represented as ns3:$_{\text{subject}}$
viii) ns3:$_{\text{subject}}$ is made into MDIU-1. This, in turn, triggers the specification that ATTENTIONAL FOCUS = 'chat'
ix) the establishment of MDIU-1 triggers ⊕ *(*DIU-*2??*MDIU-1)
x) MDIU-1 integrates with DIU-2 and yields discourse-fragment-1 since MDIU-1 was not followed by a terminator — note that MDIU-1 has not yet unified with anything in the discourse stream since it has nothing with which to unify

As before, integration of MDIU-1 into the discourse stream as discourse-fragment-1 brings in additional parameters of information: ⟨situation, intention, logical structure, semantic field, topic chain⟩. At this stage, the information is scant, as shown in (34):

(34) SITUATION: {ns3 = participant1; F-ROLE value: subject,
participant2 ??. . . , relation(s):
i) relation(s): predicate: manger(chat, souris)
time: past, punctual, complete
ii) **relation(s): ??** . . .
??. . . **time: ??**. . . }

INTENTION: ASSERTION

LOGICAL STRUCTURE: P(x . . .)

SEM: {. . . }

From the MDIU, we have the analysis shown in (35).[8]

(35) 'Il' = 'chat':

SEM: {**SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate],**
COLOUR: [blanc] . . . }

TOPIC CHAIN: chat$_{subject}$ — ??

The logical type of *assertion* is not altered because the relation of *elaboration* or *continuation* normally does not change the intentional type. All specifications are subject to corrections and/or updating later, if necessary, as illustrated in (35). Continuing with input *faim* ('hunger') yields the analysis shown in (36).

(36) NAME: ⟨**faim**⟩

CATEGORY: noun

INDEX: gen[**+fem**], num[**+sg**], pers[**3rd**]

STRUCTURE-TYPE: type: **nominal structure**

STRUCTURE-TYPE-HEAD? **yes**: HEAD = **'faim'**

F-ROLE: ?

SEM: {**[+physical condition], [+causative]: → vouloir**
(manger(subject, nourriture)) . . . }

When *faim* ('hunger') is processed, ns4 is opened and vs2 is closed, assigned the functional role 'predicate', made into MDIU-2 and integrated into the discourse stream. Since it can bind with ns3, which assumes the functional role of subject and unifies with variable ns1 in "avoir(ns3:[+animate], . . . , {ns2, compl})", it can be unified with the discourse stream and becomes discourse-fragment-2. As before, this discourse stream can be schematized as shown in (37).

(37) D = { discourse-fragment-1 ⊕ discourse-fragment-2 ⊕ ??}

With the integration of MDIU-2 into the discourse stream, we attain the situation presented in (38).

(38) SITUATION: ns3 = participant1; F-ROLE value: subject
participant2 ??. . .
relation(s): **predicate: avoir(participant3 = ns1, ns4)**
time: **past, durative**
INTENTION: ASSERTION:ELABORATION
LOGICAL STRUCTURE: P(x . . .)
SEM: {**'chat'**:
SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate],
COLOUR: [blanc]}
TOPIC CHAIN: chat$_{subject}$ — **avoir(chat$_{subject}$, ??**. . . **)**

We now complete this short discourse with the terminator ('fullstop/period'). As before during the assembly of DIU-1, when the terminator is processed, several processes are triggered:

i) ns4 is closed, assigned the functional role of direct object, made into MDIU-3, integrated into the discourse stream and because ns4 is feature-compatible with ns2 in "manger(ns1: [+animate], . . . , ns2, {compl})", ns4 unifies with ns2. At this point, ns4 fully integrates as discourse-fragment-3. In terms of the rest of the information, we now have the analysis presented in (39).

(39) SITUATION: ns3 = participant1; F-ROLE value: subject
participant2 = ns4: **F-ROLE value: direct object**
relation(s): predicate: avoir(participant1, **participant2**)
time: past, durative
INTENTION: STATEMENT:ELABORATION
LOGICAL STRUCTURE: P(x)
SEM: 'chat': {**avoir(chat, faim)**
SIZE: <normal$_{HEAD}$, [+animal], [+feline], [+domestic], [+animate],
COLOUR: [blanc] . . . }
SEM: 'faim'
{[+physical condition], [+causative]: → vouloir (manger(subject,
nourriture)) . . . }
TOPIC CHAIN: chat$_{subject}$ — avoir(chat$_{subject}$, faim$_{direct\ object}$)

ii) the current discourse-fragment chain (Il) ⊕ (avait) ⊕ (faim) becomes DIU-2. This adds the additional information parameter 'discourse-type' which is realized as ASSERTION because:
a) the intention type was ASSERTION/CONTINUATION;
b) the logical structure closed as P(x) where x = DIU-2.

iii) At this point, a number of checks are made for integrity in semantic field values, topic chain values, logical structure and discourse type. The topic chain is simple: $chat_{\text{subject}}$ — *manger*($chat_{\text{subject}}$, $souris_{\text{direct object}}$), $chat_{\text{subject}}$ — *avoir*($chat_{\text{subject}}$,$faim_{\text{direct object}}$). However, as noted above, there is a semantic link between *manger*(*chat*, *souris*) and *avoir*(*chat*, *faim*) because *avoir*(x, *faim*) usually triggers the event (*vouloir*) (*manger*(x, y)). Thus, the system can deduce that there is a cause-effect or an event-reason relationship binding DIU-1 and DIU-2. After resolving this issue, the final result will be that the discourse type of the two integrated DIUs will change from ASSERTION/CONTINUATION to NARRATION/EXPLANATION. This final set of analyses with its resultant adjustments is meant to simulate our ability and tendency to think about what we have heard or read and make necessary changes in our final impressions. These "final" impressions can always be updated and corrected later, if necessary.

Applications to CALL and Natural Language Processing

Because an implementation of DIG yields a text whose information content has been analysed and stored in a database, it becomes an excellent instrument for generating various information-based pedagogical materials and tools. This is especially useful in text analysis, paraphrasing, knowledge extraction and the preparation of various pedagogical materials, including the generation of various types of tests and quizzes. Since the information extracted covers lexical, structural, functional, contextual, situational, semantic, topical and pragmatic types of information, it becomes possible to use DIG to navigate a text in a surprising number of ways. Following is an illustrative list of possibilities:

i) A document which has been analysed by DIG would generate a database that would be able to answer numerous types of queries concerning:
   - functional roles (who? what? where? when? how? why? what kind of?, etc.)
   - word frequencies and concordances
   - conceptual summaries of texts, where chains of concepts present in texts are presented in various formats
   - flattened summaries of main information, that is at the discourse level 0 with all embedded structures removed;

ii) Exploration of problematic texts, say, to arrive at a statistical summary of types, frequency, location and density of errors or areas of ambiguities;

iii) Collecting generic information about the content of a text (extraction of words, topics, semantic concepts, etc.);

iv) Generation of vocabulary lists of various kinds including:
- words by category and immediate context
- various phrases (idiomatic, etc.)
- structures;

v) An interactive system for testing vocabulary and recall;

vi) Generating (short) subtexts with, for example, nouns blanked out in order to provide practice and reinforcement of comprehension, recall and so forth. These could be generated in a dynamic fashion, responding to the student's input as it comes in because of the incremental nature of DIG.

vii) Playing incremental word games where DIG can indicate whether an input by the student is possible at this point or not. This is a valuable exercise for mastering patterns.

viii) Generating a variety of true-false tests and quizzes;

ix) Generating and correcting comprehension tests or quizzes;

x) Generating spelling quizzes or tests based on a given text.

It is important to note that DIG studies language as an embodied natural phenomenon, existing in the real world. As such, DIG analyses will be useful for the purposes of language instruction in ways similar to useful applications of systemic linguistics. The added advantage is that DIG is a theory of incremental understanding that follows the cognitive linguistic tradition of viewing linguistics as studying psychological phenomena from a naturalistic point of view. DIG integrates into this perspective the notion of time linearity and incrementalism.

A second area where DIG can make significant contributions concerns the field of dynamic interfaces in various formal systems. The results provided by DIG indicate that a system equipped with a DIG capability could use that information to generate representations. This would require planning strategies which seem to underlie human speech production. What DIG brings to this domain of research is analogous to what it brings to the field of formal lexicon modelling: in this case, concrete indications of what is needed to keep the flow of information consistent with locally generated contexts. Ultimately, this is where DIG is heading and the goal may not be that far away.

Definition of "Information" in DIG

Although it is clear from the details presented that the lexicon plays a very important role in the modelling of information accumulation during discourse assembly, it is also equally clear that the term "information" is in itself a fairly complex concept. It can be defined as the (partial) result of acquiring values which are consistent with and integrated into the local contexts as these are built up during discourse assembly.

One of the claims behind the DIG approach is that many of the processes involved during information accumulation must operate in a semi-automatic, incremental, editable manner or else natural language understanding would not be feasible, given the severe temporal limitations imposed by real-time processing. Just how much semi-automaticity in the understanding process is due to lexical properties and how much is due to recognition of patterns and situations is a still an open question, but results achieved so far seem to indicate that when humans do master patterns or situations they are doing so in terms of a number of networked attributes which together combine to yield discourse information.

Conclusion

This paper has sketched an approach — DIG — to the study of natural language understanding, where the explanatory mechanism is the incremental assembly of representations of linguistic information through time. A representation is assembled, based on the information contained in the current word in the string under analysis, which is subsequently enriched through incorporating information collected from integrating the next word. The worked example, which provided a demonstration of the functioning of DIG, represents an application of this approach to two very simple declarative sentences. However, the use of discourse-defined unit types rather than constituency-based unit types, as well as a procedural feature-unification and feature-matching formalism promises that this approach can be scaled up to serve as a model for the description of larger, discursive examples. The theory has also been applied to compare translations and the underlying interpretive mechanisms that differentiate them from each other (Sévigny, 2003). The prospect of a linear, temporally sensitive grammatical theory explaining generative mechanisms underlying textual interpretation seems to be a natural progression for linguistic theorists who accept the notion that grammar is a cognitive and/or biological phenomenon.

A remaining question, raised by many linear grammarians, is a re-evaluation of the relationship between competence and performance that a linear approach to grammatical analysis entails. What constitutes a well-formed representation in a linear left-to-right grammar is whether there is potential progression from an initial information state to some final information state, represented by some unit (in DIG, an MDIU or a DIU). This is a radical departure from the traditional phrase-structural definitions of wellformedness. This definition may prove very useful to linguists working on second language acquisition and the construction of innovative, interactive pedagogical materials. Several of the very practical and material applications of left-to-right grammar were briefly introduced. The more philosophical and theoretical questions pertaining to the import of this sort

of linguistic theory for the study of second language acquisition and teaching promise to be the subject of many interesting debates in the near future.

Notes

I would like to thank Mike Kliffer and Lars Wessman as well as two anonymous reviewers and the English editor of this journal for their criticisms, advice and encouragement. Any errors or inconsistencies are my own.

[1] Of course, in other languages different punctuation marks might be required. In spoken language, intonation and stress phenomena would accomplish these tasks. For more detailed descriptions of complete and partial closure as well as the case of embedded structures, see The Basic Mechanisms.

[2] The concept of situation used here is less formally defined here than in Situation Semantics (Barwise and Perry, 1983); later versions of DIG may explore this relationship more fully.

[3] For details, see Kempson *et al*. (2001, Chapter 9); Marten (2002, Chapter 2).

[4] There are two sizes of the < being used in this presentation: the matched set of ⟨ ⟩ enclose NAME entries; the single opening < is intended to be read as 'less than'.

[5] See Sévigny (2000) for a more comprehensive list and description of common structure types. Because the expressions "nominal structure-x" and "verb structure-x" occur so frequently they will heceforth be abbreviated to $\mathsf{ns}x$ and $\mathsf{vs}x$.

[6] New information triggered by new input is listed in **bold**.

[7] See n. 4 for explanation regarding differences between the ⟨ ⟩ matched angle brackets and the < symbol.

[8] The bolded information here is new, not when viewed as connected to *chat*, but when viewed as being connected with *Il*, resulting from resolution of the anaphoric link *Il* = *chat*. DIG accumulates information on many levels, including structural, functional, anaphoric and semantic. Thus, information is often seen as "new" when it is being linked to a new association. This is meant to simulate the several forms of insights we experience during the process of incremental information accumulation, which occurs continuously as we process information via natural language processing.

References

Barwise, J. and J. Perry. 1983. *Situation Semantics*. Stanford, CA: CSLI Publications.

Blackburn, P. and W. Meyer-Viol. 1994. "Linguistics, Logic and Finite Trees." *Bulletin of the Interest Group in Pure and Applied Logics*, 2, pp. 3–31.

Bresnan, J. 2002. *Lexical-Functional Syntax*. Oxford: Blackwell.

Cann, R., R. Kempson and M. Otsuka. 2002. "On left and right dislocation: A dynamic perspective." Ms., King's College London.

Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Gabbay, D. 1996. *Labelled Deductive Systems*. Oxford: Oxford University Press.

Gelbukh, A. 1999. "Review of *Introduction to Computational Linguistics: Man-Machine Communication in Natural Language* by Roland Hausser." *Computational Linguistics*, 26, pp. 449–455.

Hausser, R. 1996. A Database Interpretation of Natural Language. *Korean Journal of Linguistics* 21(1), pp. 29–55.

Hausser, R. 1999. *Foundations of Computational Linguistics*. New York: Springer.

Kempson, R., W. Meyer-Viol and D. Gabbay. 2001. *Dynamic Syntax: The Flow of Natural Language Understanding*. London: Blackwell.

Marten, L. 2002. *Verbal Underspecification*. Oxford: Oxford University Press.

Martin, P. and A. Sévigny. 2000. "Intégrer une composante phonétique dans le système AIDA." Paper presented at the Annual Meeting of the Canadian Linguistic Association, University of Alberta.

Milward, D. 1994. "Dynamic Dependency Grammar." *Linguistics and Philosophy*, 17, pp. 561–605.

Pollard C. and I. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.

Sévigny, A. 2000. Lexically-Driven Incremental Discourse Assembly. Ph.D. dissertation, University of Toronto.

Sévigny, A. 2002. "Discourse Information Grammar." *Linguistics Association of Korea Journal*, 10(4), pp. 65–91.

Sévigny, A. 2003. "Information flow in excerpts of two translations of *Madame Bovary*." *Linguistica Antverpiensia N.S*, 1, pp. 100–112.

Shieber, S. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Stanford, CA: CSLI Publications.

Shin, G. 1987. Linearized Phrase Structure Grammar. Ph.D. dissertation, Chonnam National University, Kwangju, South Korea.

Sperber, D. and D. Wilson. 1995. *Relevance: Communication and Cognition*. 2nd ed. London: Blackwell.

Tugwell, D. 1998. Dynamic Syntax. Ph.D. dissertation, University of Edinburgh.

Appendix: Basic Operators

Basic Operators	Represented by	Action
lexicalize-input	lex(arg)	locates arg in the lexicon and attaches lexical attribute information template to arg
open-structure	[(arg)	reads value of S-TYPE for arg, initiates structure of type S-TYPE and marks STATE atrribute: open
concatenation	+	uses unification to build up the current structure by unifying attribute features until incompatibility and/or closure occur(s). At this point, either *close-structure* or *pause* will be triggered.
close-structure	]( )	stops unification of attribute-features with head of current structure, marks *state* as closed, triggers [( ) to initiate new current structure and finally triggers *check-adjacency-condition* (old-structure, new-structure).
check-adjacency-condition	check-adjacency-condition(arg_n, arg_{n+1})	checks the S-TYPE attribute values of arg_n and arg_{n+1}; if the case is uniquely solvable, assigns F-ROLE value to arg_n; else if the case is solvable but not uniquely, assembles all F-ROLE value possibilities in a reduced F-SET and attaches to arg_n; else the case is incompatible and processing stops
functional-role assignment	FRA(S-TYPE), where S-TYPE denotes the "type" of the structure	attaches F-ROLE value(s) to structures. In addition, if the structure is uniquely functionalized, FRA( ) assigns the status of MDIU to the functionalized structure.
integration	⊕(arg, discourse stream)	This is an inter-structural process. It requires that its first argument be functionally specified uniquely. It then follows several steps: i. effects unification of arg and discourse stream. If this is the first MDIU, it unifies trivially with the default empty discourse stream and '[( )' is applied to the new discourse stream which is now initiated and its STATE value is marked "active"; ii. if there is a terminator, triggers *complete-closure*( ) (denoted as '##') and assigns discourse stream the status of DIU; iii. resumes processing
complete-closure	##	recursively closes all structures and adjusts discourse levels. Also, if all structures close successfully, *complete-closure* collates the MDIUs into a single DIU and generates the information relevant to it. If all structures do not close, this operation stores the information, lists open structures and signals discourse closure error