11 COMPUTATIONAL MODELS OF INCREMENTAL GRAMMATICAL ENCODING

Koenraad de Smedt

Chapter prepared for:
De Smedt, K. (1996). Computional models of incremental grammatical encoding. In A. Dijkstra & K. de Smedt (Eds.) (1996). *Computational psycholinguistics: AI and connectionist models of human language processing* (pp. 279-307). London: Taylor & Francis, 1996.

11.1 Introduction

In formulating a sentence, speakers must take into account not only what they want to say and the context in which they are saying it, but also the grammar of the language they are speaking. *Grammatical encoding* is the casting of a message into a linguistic plan that conforms to the grammar of a language. Together with *phonological encoding*, it is part of a module called the *Formulator* which occupies a central place in the blueprint of the speaker drawn by Levelt (1989, p. 9; see also Chapter 1). The input to the Formulator consists of preverbal messages generated by the *Conceptualizer*. I will assume that this input is given as *conceptual structures,* without making detailed assumptions about their form, except that they consist of conceptual entities and relations. The output of the Formulator is a phonetic plan that can control the articulation processes.

Grammatical encoding comprises both the selection of lexical material and the organization of this material in a syntactic framework. The present chapter will concentrate on the latter aspect, while lexical selection is dealt with in Chapter 12. The assembly of a syntactic framework involves structural choices that are subject to various restrictions imposed by the grammar but are also sensitive to various performance factors which presumably originate in situational and architectural characteristics of the speaker's cognitive system. Some of these factors will be explained in Section 11.2. But first, I will briefly touch upon three grammatical restrictions which play a role in this chapter: subcategorisation, word order, and the use of markers such as inflection and function words.

First, the syntactic form of sentences is to some extent determined by certain characteristics of the chosen words. More specifically, it is assumed that words are *categorized* in the lexicon according to what parts of speech they are. The English word *dream,* for example, is categorized as a verb or a noun, but it cannot be used as an article nor as a preposition. Furthermore, words are *subcategorized* according to the way they can be complemented to form a complete utterance. For example, certain verbs take *that*-clauses as direct objects, as in *I dreamt that I could fly,* whereas others take a *to-infinitive* instead, as in *I wanted to fly*. This subcategorization is not predictable from the meaning, so that an interdependence between word choice and syntactic planning must be assumed.

Second, spoken or written utterances are necessarily linear in form. Words and phrases cannot always be put in the same order as that in which the corresponding ideas have come to the speaker's mind. In English, word order is rather fixed and canonical linear patterns may convey meaningful distinctions. The question *Was she late?* has a different word order than the assertion *She was late*. Other variations may be stylistically marked: *Late she was,* or may simple be wrong: *She late was*. Still, the syntax of English allows some freedom of choice in the left-to-right order of words. We can say both *Yesterday she was late* or *She was late yesterday*. Consequently, we can ask the question how such choices are accounted for in a performance model of sentence production.

Third, markers such as inflexion (see also Chapter 7) and the use of function words, including articles, prepositions, and auxiliaries, are prescribed by the grammar to express certain aspects of meaning. For example, the feature *plurality* is normally expressed in

English by a suffix that makes the difference between *horse* and *horses,* and the feature *definiteness* is marked with an article in *the horses*. The use of these markers is governed by rules which seem arbitrary and language-dependent. Russian, for example, has no articles at all. Moreover, the use of these markers is dependent on other parts of the planned utterance. English has a rule for *subject-verb agreement* (SVA) which requires finite verbs to agree in person and number with their subjects: *He walks* vs. *They walk*. The computation of agreement presupposes knowledge about grammatical functions, such as subject, which the speaker must somehow identify in the sentence plan.

Summing up, planning an utterance involves choices that take into account several grammatical restrictions, of which I have mentioned word order, the use of inflexion and function words, and subcategorization. These restrictions generally presuppose structural knowledge about word groups (such as noun phrases) rather than single words. Moreover, word groups and sentences can involve several levels of embedding. Language production therefore seems to require representations and processes that manipulate hierarchically organized syntactic units. It is generally accepted that sentence planning involves hierarchical representations, usually depicted in the form of tree shaped graphs (see also Chapter 8), and yields a frame with ordered slots for words. The structural knowledge expressed in a graph allows features like gender, number, etc., to be assigned to words where appropriate, in order to allow the computation of inflected phonological forms at a subsequent level of the language production process.

Given a conceptual input, a grammar, and a lexicon, the speaker has all the necessary knowledge to start building the linguistic form of an utterance. But how does the speaker go about it? From a computational viewpoint, grammatical encoding can be viewed as the satisfaction of multiple constraints. The final utterance must be both well-formed according to the grammar and composed of existing words that fit in the syntactic structure. It must, of course, also express the conceptual input as precisely as possible. Choices on one level may conflict with choices on another level. Furthermore, all the information from the various sources of knowledge may not necessarily be accessible at the same moment. The process needs to cope with certain limits of its computational resources, for example, the size and extent of working memory. Consequently, it should not be surprising that errors and hesitations are quite common in on-line speech production. Still, the language production process generally succeeds in the timely generation of utterances that convey the intended meaning. These high demands presuppose some amount of sophisticated planning that is supposed to proceed in an *incremental* (piecemeal) way. The goal of this chapter is to consider some computational models that provide an account of syntactic representation and processes for incremental language production.

The next section will present a selection of linguistic and psycholinguistic phenomena, leading to the formulation of some important problems and assumptions in grammatical encoding. Then the chapter will zoom in on the incremental mode of sentence production. A number of computational models which account for this processing mode will be discussed in detail. Although the models have been developed for different languages (Dutch, German or English) and sometimes aim at giving an account of specific word order phenomena in that language, a comparison can be made. This is done in the final section.

11.2 Phenomena and assumptions in grammatical encoding

What clues can we find for a psychologically plausible architecture for grammatical encoding? Some indications can be found in studies of speech errors and dysfluencies. Even if spontaneous speech generally appears to be uttered relatively smoothly, closer inspection reveals some performance problems. Speakers make short pauses, repeat themselves, make mistakes, correct themselves, stop in mid-sentence, start all over, etc. Some of these phenomena can be seen as indicating the difficulties and limitations of syntactic processing. Other clues can be found in regular variations in the forms of utterances. Even if the grammar prescribes the linguistic form of our utterances, it also allows certain degrees of freedom. For example, we can choose between active and passive sentences, or between alternative word orderings. Such choices are neither entirely random, nor completely predictable from the underlying message or the context of the utterance, but seem partly to reflect certain performance aspects. Put briefly, empirical work suggests that variations in the forms of utterances, as well as errors and hesitations, commonly reflect variations in the processing of linguistic information (for example, Bock, 1982; 1986; Bock & Warren, 1985; Garrett, 1980; Stemberger, 1985). Some of the processing factors which may exert such influences are discussed below. For a wider overview of observational and experimental evidence about the syntactic component in speech planning, I refer to Bock and Levelt (1994) and Bock (in press, a).

11.2.1 Conceptual and lexical guidance

If one hypothesizes the need for syntactic plans in sentence production, then one of the main questions is how to co-ordinate word selection and syntactic planning. There are occasions when the production process may be driven by syntactic patterns which follow directly from the content and structure of the message to be transferred. A definition, for example, is likely to be built on the syntactic pattern *An x is a y that z.* In addition, there is evidence for correspondences between the meaning of a verb and its subcategorization (Fisher, Gleitman & Gleitman, 1991). Whether a verb is transitive or not, for example, is somewhat predictable from the kind of action it expresses. This seems to suggest that syntactic plans can be created on the basis of the meaning to be expressed. However, if this is done without regard for the specific words to be inserted, this would still give rise to ungrammaticalities. Consider the sentence patterns in (1), which are constrained by the different subcategorizations of the verbs *replace* and *substitute,* even if these verbs are very similar in meaning.

(1) a. Cecile replaced / *substituted Greek literature with Spanish.

b. Cecile *replaced / substituted Spanish literature for Greek.

It makes sense to assume that lexical knowledge guides the construction of a syntactic structure (Kempen & Hoenkamp, 1987). This guidance could be accomplished by *lemmas,* lexical entries associating meaning with words in an abstract syntactic plan (see Chapter 12). For example, there is a lemma associating the verb *substitute* with its meaning and with a syntactic plan where it is followed by a direct object (expressing the replacement) and a prepositional clause with *for* (expressing the replaced item).

While lexical choice thus seems to determine the syntactic structure, it seems reasonable to assume that lexical choice is at the same time constrained by the given syntactic context. Speech errors, in particular blends or fusion errors, suggest that lemmas may compete during grammatical encoding. In utterance (2), for example, it seems that *understanding* and *compassion* are both available, but it is not possible to integrate both in the current syntactic structure.

(2) ...that one would get understanding for this compa … uh … compassion for this character.[1]

The notion of the lemma is complemented by the notion of the *lexeme,* which defines a particular phonological word form suitable for a given syntactic context. The separation of semantic-syntactic information in lemmas on the one hand and morpho-phonological information in lexemes on the other hand has been assumed by various authors who suppose that lexical access is a two-stage process (Garrett, 1980; Levelt & Maassen, 1981; Kempen & Huijbers, 1983; Levelt & Schriefers, 1987). The first stage is the selection of lemmas (see Chapter 12) which are configured into a syntactic structure during grammatical encoding (present chapter); the second stage is the retrieval of lexemes during phonological encoding (see Chapter 13).

11.2.2 Performance factors in word order and syntactic choice

Incremental production

Spontaneous speech is to some extent planned *incrementally*, that is, in a piecemeal way. According to a stage model as outlined in Chapter 1, a message must be delineated before lemmas for it can be retrieved and before it can be grammatically encoded. But this kind of seriality does not necessarily apply to the whole sentence. It would be odd if speakers always first conceptualized the complete content for the next sentence, then retrieved all lemmas, then created a syntactic structure, next computed the phonetic string for the whole utterance and only then started uttering the first word. Instead, there is evidence that sentences are not always planned in their entirety before speakers start to articulate them. Experimental work by Lindsley (1975, 1976), Levelt and Maassen (1981) and Kempen & Huybers (1983) suggest that while there is some planning ahead, speakers start articulating their utterance not long after the first words (but not yet the whole sentence) can be produced. This is in line with some earlier studies on hesitations in speech (e.g. Boomer, 1965; Goldman-Eisler, 1968; Brotherton, 1979).

More recent empirical work with picture description tasks has focused on the effect of *conceptual* and *lexical accessibility* on incremental production (Bock & Warren, 1985; Kelly, Bock & Keil, 1986; McDonald, Bock & Kelly, 1993). The hypothesis is that variations in the syntactic structure of sentences, other things being equal, are partly determined by the ease of retrieving the words' conceptual content and the lemmas to express this content. Among other things, assignments of grammatical functions, such as subject and direct object, would be affected by the ease of finding a lexical representation consistent with the intended meaning. Similarly, Pechmann (1989; Schriefers & Pechmann, 1988; but see Pechmann, 1994) hypothesised that variations of word order, e.g., *a white big triangle* rather than the preferred *a big white triangle,* will be partly determined by the ease of accessing the words' conceptual content. Experimental results point in this direction, although the evidence is not conclusive.

There are several ways of viewing incremental production. One way allows various stages of sentence production to run in parallel by letting one module start processing before the previous one has completely finished. This approach, called *pipelining* in computer science, was first proposed for sentence production by Kempen and Hoenkamp (1987) and has been implemented in a number of computer models discussed below in section 3.1. Other models are not pipelined, but are incremental in the sense that small amounts of work is being done in successive small amounts of time. One model, discussed below in section 3.2, lets grammatical encoding proceed stepwise by choosing successive words to be incorporated in the utterance one by one, while all conceptual inputs and all linguistic knowledge act together.

Syntactic weight

Ross (1967) coined the term 'heavy-NP shift' to describe the phenomenon in English where a long or *heavy* noun phrase (NP) tends to appear in sentence-final position, as in (1a). Hawkins (in press) found that the length of the NP (counted in words in written English) is less important than its length relative to other constituents which could occupy sentence-final position, for example the prepositional phrase (PP) *to Claire* in (3).

(3) a. I introduced to Claire *some friends that Alex had brought to the party*.

b. I introduced *some friends that Alex had brought to the party* to Claire.

More specifically, at equal weight there is a clear preference for the normal NP PP order in (3b), but as the relative NP weight increases, this preference diminishes and NP movement to final position becomes more likely. De Haan and Van Hout (1986) have found similar *weight* phenomena in a statistical analysis of postmodifying clauses, including relative clauses, in a corpus of written English.

Hawkins (1990) proposes an account of weight phenomena in terms of the ratio of immediate constituents (ICs) to words in a sentence. The assumption is that having longer constituents toward the end improves the IC-to-word ratio early in the sentence. This is viewed as an advantage for sentence comprehension. However, it hardly explains the phenomenon from the viewpoint of the speaker. On-line experiments by Stallings, MacDonald and O'Seaghdha (1994) suggest that heavy-NP shift facilitate both comprehension and on-line production.

No definitive explanation has been proposed so far. Assuming that heavy constituents place extra demands on processing and memory resources, they may be 'overtaken' by competing constituents which are lighter, or they may be postponed to a point in the utterance where overall processing demands are low. However, processing preferences directly based on surface features such as relative length seem difficult to integrate in a theory of incremental production, because the length of a constituent may not yet be known when its utterance is initiated. Alternatively, heavy constituents may be expressing less accessible complex concepts.

Persistence

Syntactic persistence is the spontaneous repeated use of a particular syntactic construction across successive sentences. This effect has been reported in natural conversation (Kempen, 1977) but has also been found in a controlled context. Priming experiments suggest that the choice of a particular syntactic plan can be positively biased by prior activation of an instance of such a plan. Bock (1986) conducted experiments with a picture description task where subjects who first produced a prime sentence with a given structure were asked to describe a picture of a conceptually unrelated event. Subjects who had just produced the prepositional dative sentence (4a) were more likely to describe a subsequently shown picture as (4b) than by using the double-object dative (4c). The opposite pattern was also found for the double-object dative (4d) as a prime.

(4) a. The governess made a pot of tea for the princess.

b. A clerk is showing a dress to a man.

c. A clerk is showing a man a dress.

d. The governess made the princess a pot of tea.

A similar tendency was observed for active vs. passive sentences. These findings suggest not only that there is competition between alternative structural realisations, but also that recently used structures are at an advantage in this competition. Recently used structures may be more *activated* than competing but less recently used structures (see Chapter 3 for an explanation of the activation metaphor, and Chapter 8 for the use of activation in models of syntactic analysis).

Syntactic coherence and repairs

Errors affecting the *correctness* and *syntactic coherence* of the utterance occur relatively often in spontaneous speech. Word exchange errors, as in (5), provide the basis for an argument that different parts of the sentence are simultaneously present during sentence planning (Garrett, 1980). In computational terms, this argues for allowing different branches of the syntactic tree structure to be constructed in parallel (cf. 11.3.1).

(5) We all sing around and ... stand around and sing.

One type of syntactic incoherence is the *apokoinou*, where the middle of a sentence goes with both the first and the last part of a sentence. For example, in (6a), the first part, between braces, is in itself grammatical, and so is the overlapping final part between square brackets; however, the sentence as a whole is incoherent. Such utterances are sometimes treated as *non-retracing repairs* (De Smedt & Kempen, 1987), because they can be seen as a kind of repair that is uttered without a correction marker and often without hesitation, in contrast to syntactic rearrangements which appear as normal retracing repairs (6b).

(6) a. {So we get as [an added bonus to this approach} is a system which …]

b. It smells a little … a little citrusy is what it smells like.

Spontaneous apokoinous might be explained in computational terms as the intrusion of competing syntactic frames into a syntactic structure of which the less recent elements have 'decayed'. The most recently uttered constituent is used as a hook to attach a new sentence pattern. The phenomenon could also be seen as evidence for the simultaneous exploration of several syntactic structures. However, even if they imply the construction of several parallel structures, the various alternative structures need not be active at the same point in time, but could be active in succession, as already suggested above by their treatment as non-retracing repairs.

11.2.3 Structural dependency in subject-verb agreement

The morphology of a word may be structurally dependent on features of other words. In *subject-verb agreement* (SVA) in English, the inflexion of the verb is under the control of the features of the subject noun, in particular its features number and person. How do speakers implement dependencies that span multiple words or constituents? The presence of a noun with diacritical parameters different from the subject, in a position close to the verb, may promote errors against SVA, as in (7).

(7) a. * The producer of the latest films were in financial trouble.

b. * The producer who made the films were in financial trouble.

Such errors have been labelled *attraction errors* or *proximity concord* by some authors (for example, Quirk, Greenbaum, Leech & Svartvik, 1972), reflecting the fact that the more recent noun plays a role as an attractor. Recency in itself, however, does not explain the fact that SVA errors are more frequent across *phrases* (7a) than across *clauses* (7b), as found in error eliciting experiments (Bock & Miller, 1991; Bock & Cutting, 1992; Bock, in press, b). It was further found that only in the phrase condition did the error frequency increase with phrase length, which means that processes responsible for agreement are most sensitive to the diacritical parameters of noun phrases within the same clause. Bock and Cutting conclude that the ability to maintain information in working memory seems necessary to implement dependencies that span multiple words or constituents, and, more importantly, sentence production seems to rely on a hierarchical architecture where clauses are separate planning units. Vigliocco, Butterworth & Semenza (in press) share this view but argue against an autonomous syntactic module because they found morpho-phonological and semantic effects on SVA errors in several languages. Furthermore, an asymmetry between singular and plural has often been found.

11.3 AI-based models

The previous section briefly reviewed some important phenomena and assumptions with respect to grammatical encoding. We now turn our attention to computational models. A wealth of implemented computer models for sentence production has emerged from the Artificial Intelligence, Computational Linguistics and Cognitive Science traditions.[2] Some of these models are hardly or not at all related to psycholinguistic insights into sentence production. No known computer model takes into account all of the phenomena described above, let alone the remaining phenomena in the literature that are not discussed here. Many computational models, however, do address the incrementality issue explicitly. Several incremental models based on an AI paradigm will be discussed in detail in this section.

11.3.1 IPG

Incremental Procedural Grammar (IPG; Kempen & Hoenkamp, 1987; Levelt, 1989) is a computational model for sentence production which claims that human sentence production involves parallel activity in the different modules responsible for *conceptualising, formulating* and *articulating*. Each of these modules takes the output of the previous one as its input, but the modules can be organized so that they operate on different pieces of input concurrently, as illustrated in Figure 11.1. The scheme shows *intercomponent* parallelism, i.e. the simultaneous activity of the various modules. This allows the Formulator to begin producing a sentence before the Conceptualizer has completely defined all of its content. In addition, the overlapping lines in the Formulator stage show *intracomponent* parallelism, i.e. the simultaneous processing of fragments within a module. This allows several branches of the syntactic structure to be computed in parallel.

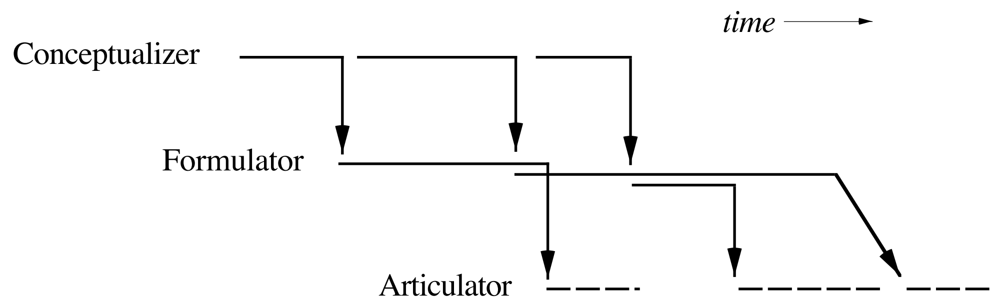


Figure 11.1 Schematic view of incremental generation in IPG.

IPG is a model with a *procedural* architecture (see Chapter 2), implemented on a computer and tested with a grammar for Dutch. Syntactic planning is executed by procedures which in turn call other procedures to do part of their job. A distinction is made between categorial procedures, which plan syntactic categories such as a noun phrase (NP), a noun (N), a sentence (S), a verb (V), etc., and functional procedures, which plan syntactic functions such as subject (Subj), direct object (Obj), etc. IPG claims that the choice of these procedures is largely controlled by the grammatical properties of the lemmas which are retrieved to express a given meaning. As an example, consider the Dutch lemmas for *zien* (see) and *willen* (want), shown in (8a,b). Both are verb (V) lemmas which require slots for a subject (Subj) and a direct object (Obj). Examples (8c,d) are instances of nominal (N) lemmas.

(8) a. V (nil, <Lexeme (zien)>)
Subj (Path (actor:), <>)
Obj (Path (object:), <>)

b. V (nil, <Lexeme (willen)>)
Subj (Path (actor:), <>)
Obj (Path (object:), <>)
ObjComp1

c. N (nil, <Lexeme (Chris)>)

d. N (nil, <Lexeme (film)>)

Each lexical specification consists of a conceptual pointer (possibly *nil*), and lexeme (between angled brackets, possibly empty). The specifications in the verbal lemmas cause calls to the dedicated subprocedures named V, Subj and Obj. They also indicate what parts of the message are to be associated with the Subj and Obj procedures. More generally, syntactic procedures are always specialized according to the kind of constituent they produce, and they operate within the constraints that the lemmas impose on their syntactic and conceptual environment. An example of a lexically specified syntactic constraint is the feature ObjComp1 in (8b), which signifies that the object complement is an infinitival clause.

As a simple example of how incremental production proceeds, consider the generation of the Dutch sentence (9). The conceptual structure serving as input consists of conceptual entities representing objects, actions, events, etc. in the message to be expressed, linked by conceptual relations, sometimes called *case relations* or *thematic roles*, for example *actor, possessor, location,* etc. This structure is referenced by a conceptual pointer (*cp*) as shown in (10). Furthermore, suppose that the conceptual structure is not accessible all at once, but that initially only the concept in (10a) is accessible, and at successive time intervals the expansions in (10b,c) become available.

(9) Chris heeft de film willen zien. (Chris has wanted to see the film)

(10) a. cp = Chris

b. cp = want (actor: Chris) (aspect: perfect)

c. cp = want (actor: Chris) (object: see (actor: Chris) (object: film))

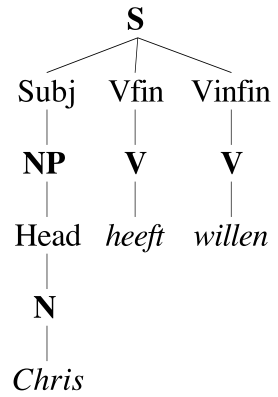
To start the grammatical encoding of a sentence, an initial procedure for the main sentence (S) is set up with the appropriate conceptual pointer as its argument, as in (11).

(11) S (cp, <Main>)

IPG presupposes that a lemma is selected that satisfies the conceptual structure (see Chapter 12 for models of lemma selection). Lemma (8c) satisfies the conceptual specifications in (10a). This lemma gives rise to a categorial procedure N which attempts to become active in the context of S. However, N cannot be directly called by S, because it first needs to be assigned a grammatical function. According to the grammar, N can be the head of an NP; consequently, a categorial procedure NP is created that will call a functional procedure Head, which will in turn call N. Furthermore, an NP is by default the subject of an S, so a functional procedure Subj is created. This gives rise to the initial procedure calling hierarchy in (12).

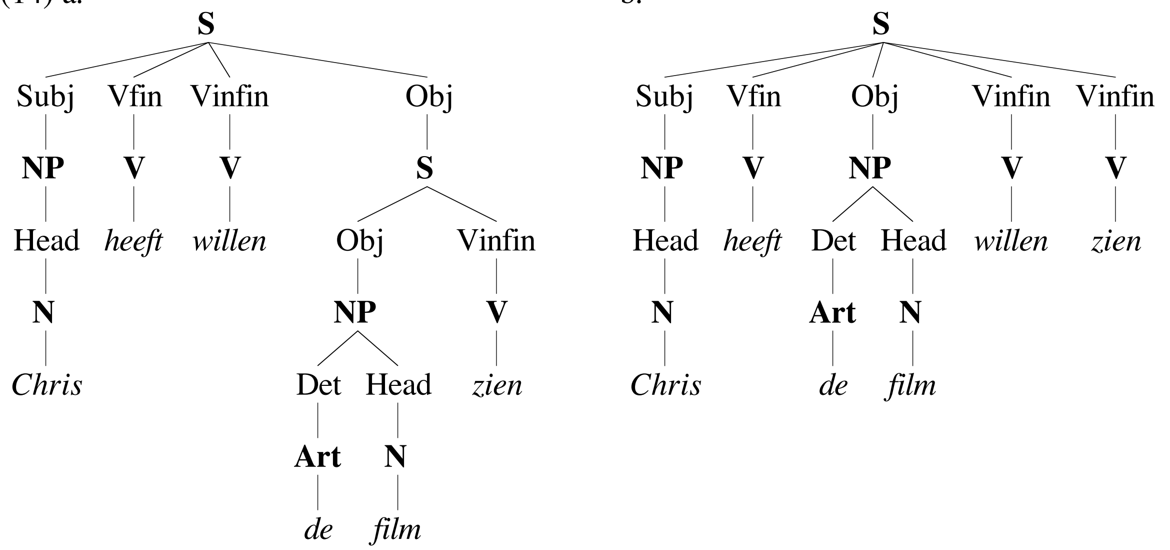
(12)

(13)



IPG uses iteration as a basic procedural mechanism to model incrementality. Each procedure repeatedly checks for the next conceptual expansion and processes it. Suppose, for example, that the next expansion of the conceptual structure is that in (10b). Suppose furthermore that lemma (8b) is retrieved and creates the necessary subprocedures. Subj is a functional procedure which can be called directly by S. Moreover, it is already present, and its conceptual structure is consistent with the *(actor:)* specification in the lemma. The perfect aspect gives rise to the addition of a procedure V for an (auxiliary) verb. V is a categorial procedure and must first be assigned a grammatical function. In this case, V is a finite verb, and therefore a procedure VFin is created. Another V procedure for the main verb is spawned and it is assigned the grammatical function Vinfin because it is a non-finite verb. The current state is reflected in (13). In the final iteration, the *cp* in (10c) becomes accessible. Skipping further details, structure (14a) represents the eventual results of expansion of the procedure call hierarchy with an Obj branch. This structure is still unordered.

(14) a. b.



Now let us consider IPG's treatment of word order, and in particular the discontinuity in (9) which has not been explained so far: The verb *willen* apparently breaks up the subclause *de film zien*. In IPG, each subprocedure computes a value as its output which represents the syntactic constituent that has been produced, and normally this value is returned to the calling procedure. Categorial procedures determine the left-to-right position of the constituents they receive from the functional procedures. As a result, subclauses and other hierarchically embedded structures are kept together in the linear order of the resulting sentence. But the model also contains mechanisms for a value return that is exceptional, as for the discontinuity at hand.

This exception is due to the influence of a piece of information coded as ObjComp1 in lemma (8b) which allows for an infinitival clause. Such clauses may exhibit the kind of discontinuities mentioned, which are commonly known as *clause union*. The exceptional lexical information responsible for this structure specifies that the constituents of the lower S are to be returned to the next higher S. The positions assigned to the *raised* constituents by the higher S break up their grouping, so that the discontinuity arises, as illustrated in (14b). IPG implements this value return mechanism by using the scope of variables in the procedure hierarchy.

The incremental mode of generation in IPG has its consequences for word order. Categorial procedures assign newly constructed material a position as early in the utterance as possible within limits of grammaticality. A sentence may, for example, start out with *Yesterday…* if that part of the conceptual structure underlying this word is the first which is accessible. This initial utterance might be completed as *Yesterday … Chris has seen the film*. In different circumstances, the concept underlying *yesterday* may become accessible much later, resulting in *Chris has seen the film … yesterday*. Clearly, each iteration of the looping procedures should take care to continue adding to a syntactically coherent sentence, rather than producing a succession of disconnected utterances. Measures to this effect include certain syntactic adaptations, including a shift to the passive voice. Suppose that the object role is accessible before the actor, then the previous example might be uttered as *The film …*

was seen by Chris yesterday. In this regime, clearly there may be occasions when early choices lead to partial utterances that cannot be completed with the newly incoming input, representing a case where the speaker has 'talked himself into a corner'. For instance, it may not be possible to continue *The film* … to express the fact that Chris has wanted to see the film.

Summing up, incremental production is modelled in IPG by expanding the conceptual structure presented as input to the Formulator at each iteration. The more accessible parts of the conceptual structure are assumed to enter the Formulator sooner and are processed during earlier iterations. They are at an advantage to be realized at an earlier position in the sentence and to take preferential grammatical functions. IPG proposes intercomponent and intracomponent parallelism, even if these forms of parallelism have only partially been implemented. The procedural approach opens up the possibility of accounting for linguistic phenomena not in terms of grammar rules but in terms of the computational properties of the syntactic processor (compare Marcus, 1980, for a similar approach to parsing). Indeed, IPG explains some grammatical constructs, including conjunction and locality constraints on raising, not so much in terms of explicit grammar rules but in terms of computational properties of the processor.

11.3.2 IPF

The Incremental Parallel Formulator (IPF; De Smedt, 1990; 1994) has taken the work in IPG several steps further, especially with respect to the timing of the input, the implementation of parallelism, and the grammar formalism. Variations in conceptual accessibility are modelled by letting conceptual *fragments* enter the Formulator one by one, rather than as cumulative expansions of a single conceptual structure. These input fragments are assumed to be small and correspond roughly to a single word or thematic role. Moreover, they enter the Formulator at precisely specified time intervals. For example, a typical input sequence for the generation of the Dutch sentence (15a) could be (15e). Note that (15b,c,d) are some of the possible variants of the sentence.

(15) a. Chris heeft vandaag de film gezien. ('Chris has today the film seen')

b. Vandaag heeft Chris de film gezien. ('Today has Chris the film seen')

c. Chris heeft de film gezien vandaag. ('Chris has the film seen today')

d. De film heeft Chris vandaag gezien. ('The film has Chris today seen')

e. concept CHRIS
(wait 5 time units)
concept SEE (+ perfective)
(wait 3 time units)
thematic role ACTOR between CHRIS and SEE
(wait 5 time units)
concept TODAY
(wait 2 time units)
thematic role MODIFIER between TODAY and SEE
(wait 4 time units)
concept FILM (+ definite)
(wait 3 time units)
thematic role OBJECT between FILM and SEE

IPF presents a detailed implementation of both intercomponent and intracomponent parallelism that goes beyond the iterative control structure of IPG. The moment when each individual conceptual fragment (either a concept or a thematic role) enters the Formulator can be precisely specified in the input by means of the *wait* instructions in (15e). For each incoming conceptual fragment, the Formulator immediately spawns a computational process that may overlap in time with any other active processes and may compete with them. Because there is no central controlling mechanism, grammatical encoding is performed in a parallel and distributed fashion. However, all processes share the same memory, so that together they build parts of a single coherent syntactic structure. IPF does not model the syntactic structure as a hierarchy of procedures, as IPG does, but rather as a heterarchy. It allows a more flexible order in which various parts of the utterance are computed. In particular, IPF takes the following requirements formulated by Kempen (1987a) seriously:

1. Because it cannot be assumed that conceptual fragments which are input to the Formulator are chronologically ordered in a particular way, it must be possible to expand syntactic structures from the bottom up as well as from the top down.
2. Because the size of each conceptual fragment is not guaranteed to cover a full clause or even a full phrase, it must be possible to attach individual branches to existing syntactic structures, including the addition of sister nodes.
3. Because the chronological order in which conceptual fragments are attached to the syntactic structure does not necessarily correspond to the linear precedence in the resulting utterance, word order should be assigned incrementally by exploiting grammatical variations on the one hand and should contain mechanisms to avoid ungrammaticality on the other hand.

The incremental grammar which Kempen (1987a) proposes has later been called Segment Grammar (SG; De Smedt & Kempen, 1991). This grammar is distributed: it consists of a set of *syntactic segments,* which are representations of separate grammatical relations to which features and word order constraints are added. By virtue of its organisation, the grammar not only specifies which sentences are grammatical, but also which *partial* structures are grammatical if continued. The constituents of a sentence with their grammatical relations (somewhat different from those in IPG) are represented in *functional structures.* An example of a functional structure and the segments from which it is made up are shown in Figure 11.2.

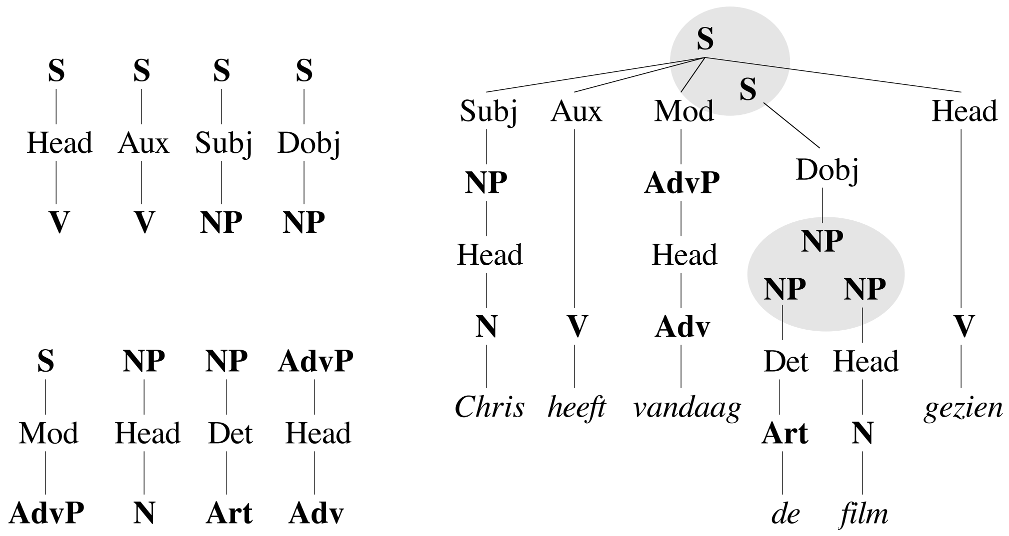


Figure 11.2 Syntactic segments and functional structure for sentence (15a).

A functional structure is formed by incrementally adding segments by means of a general *unification* operation which merges two nodes into one. Because unification is not restricted to expanding the tree in a particular direction, the tree can be expanded top-down as well as bottom-up and breadthwise as well as depthwise. As an illustration, the unifications resulting from the last two increments of (15e) are highlighted by means of a grey background in Figure 11.2. The penultimate increment selects an *NP-Head-N* segment for *film* and attaches the definite article *de* in a *NP-Det-Art* segment. Finally, the conceptual increment for the thematic role *object* causes insertion of the *S-Dobj-NP* segment.

Unification is an information combining operation proposed by Kay (1979, 1984) and is used in several grammar formalisms (see Chapter 2). Segment Grammar uses a unification operation which combines two nodes into one node having the features of both. Corresponding feature values should match. For example, a singular NP node can only unify with another NP node that is also singular. If one of the nodes is singular and the other undetermined with respect to singular or plural, the new node becomes singular. This feature matching enforces syntactic constraints such as subject-verb agreement.

Word order is computed incrementally by constructing an ordered constituent structure on the basis of the functional structure. Like IPG, the SG grammar used in IPF allows the construction of a constituent structure which is not necessarily isomorphic with the functional structure, thus providing for the incremental production of discontinuous constituency (De Smedt & Kempen, in press).

Especially important for incremental production is the way in which choices are made between alternatives in word order, as between (15a,b,c,d). In IPF these are naturally explained as the outcome of a race between constituents. Because in the input (15e), the concept for *Chris* is the first one accessible, it has a higher chance of occupying a position early in the sentence. If the input is reordered and its fragments passed on to the Formulator at different times, other word orders such as (15b,c,d) may emerge, due to a similar principle as in IPG, that constituents try to occupy the earliest position in the sentence that is still free as well as grammatically permissible. Computer simulations of IPF have demonstrated that such variation in word order indeed occurs. Assuming that the *topic* of a sentence is relatively more conceptually accessible, then the IPF processing strategy can account for the fact that the topic is often fronted (Bock, 1982; Bock & Warren, 1985; Levelt, 1989).

IPF underwrites Kempen and Hoenkamp's (1987) proposal to let an incremental mode of production trigger not only marked word order variations, but also lexico-syntactic operations such as *passivization*. Thus, for example, the passive voice in *The film has been seen by Chris* may be due to timing effects. Abb, Herweg and Lebeth (1993) focus on the incremental generation of passive sentences, and work out the assumptions of IPG and IPF in the context of the SYNPHONICS model for German sentence production. Their approach differentiates explicitly between two types of conditions that can trigger the incremental production of passives as opposed to actives.

The first condition which may give rise to passive sentences is indeed related to *prominence* and *timing*. A prominent conceptual fragment may be made available to the Formulator prior to the event in which it plays a role, even if its thematic role in that event is not yet specified. The SYNPHONICS Formulator is guided by heuristic principles including the following: "Integrate phonologically filled material as soon as possible into the highest and leftmost position available in the current utterance fragment". According to this heuristic, similar to one in IPG and IPF, the phrase expressing the prominent fragment (*the film* in our example) is inserted into the most prominent syntactic position and by default assigned the nominative case. So far, no specific information about the fragment's thematic role has been used. At a later moment, however, such information becomes available, and lemma selection is then restricted not only by the conceptual content, but also by the syntactic structure produced so far. In the present case, lemma retrieval must choose a lemma for *see* that allows the patient to be subject (because it has already been assigned the nominative case). This is exactly the property of the passive form of this lemma.

The second case is triggered by *agent backgrounding,* a condition where the agent of an action is unknown, defocused or backgrounded. In this condition, lemma retrieval results in the choice of a passivized lemma, which is subcategorized as taking over all but the first argument (which would the subject of the active voice). In this way, a passive sentence can be realized without the agent.

To summarize, IPF is a further development of IPG. In the parallel distributed architecture of IPF, incrementality is gradual, because the time span between successive inputs can be made to vary. The more time there is between successive inputs, the more time each grammatical encoding process has to 'get ahead' of the next ones and the more the effects of incremental processing will show up in the form of the utterance.

11.3.3 POPEL

POPEL (Finkler & Neumann, 1989; Neumann & Finkler, 1990; Reithinger, 1991; 1992) is a language generation system consisting of two main modules: a Conceptualizer POPEL-WHAT, and a Formulator POPEL-HOW. The system is not explicitly meant as a simulation of human language production. Rather, it is meant to be used in a human-computer interface where written dialogue is combined with other interaction modes including gestures. Still, its architecture shares the assumptions of other, psychologically motivated approaches, including incrementality, and intercomponent as well as intracomponent parallelism. This need not be surprising. On the contrary, it illustrates the point that nature inspires technology. I will restrict this section to an explanation of how the incremental generation of German topicalization constructions is realized in POPEL-HOW.

The input to the model is an intentional goal, for example, to make the hearer believe a state of affairs. The conceptual knowledge representing this state of affairs can be represented in a graph consisting of nodes representing concepts and links labelled with roles, for example the graph in Figure 11.3. The information in this graph will in fact be realized in two separate sentences.

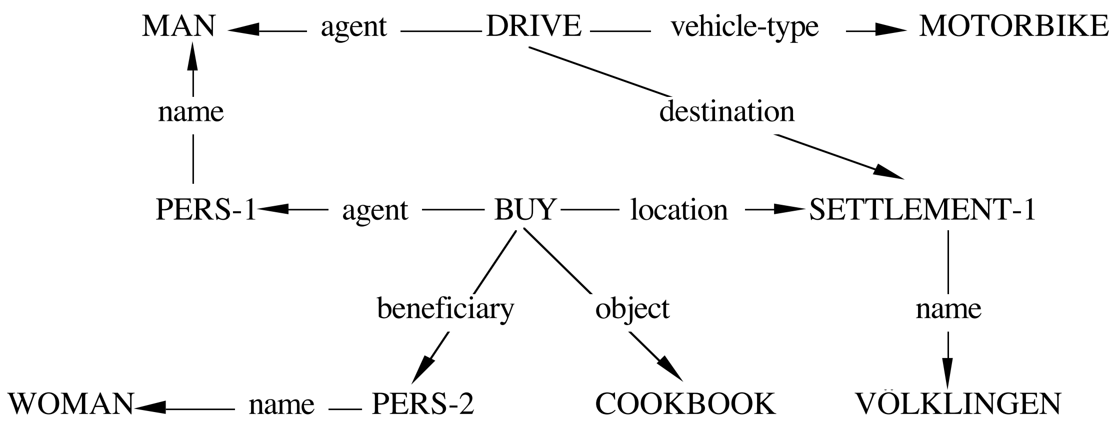


Figure 11.3 A conceptual graph in POPEL.

The Conceptualizer, which selects concepts for formulation, operates independently from the Formulator. It incorporates a user model and a dialogue model, among other things, to guide its selection of what to formulate next. The selection process has knowledge about standard schemata to structure a message (see Chapter 10). However, if the actual dialogue requires cohesion in a text (see Chapter 10), the default selection sequence may be overridden, and conceptual elements that connect to the previous discourse are selected and activated first. They trickle down the cascade, and if the grammar rules allow for it, are verbalized first.

Take as an example the incremental generation of the sentences in (16) and (17), which together express the knowledge in the graph of Figure 11.3. There is no marker in the whole system that says that a particular concept is in focus and has to be put in front. Rather, the surface position and form is dependent on the prominence of the concept, as apparent from its position in the activation cascade. That is, concepts which are most prominent and reside in immediate memory are put first in the cascade. For instance, the word *dort* is prominent in (17), uttered immediately after (16), where its referent *Völklingen* has just been mentioned.

(16) a. Ein Mann fährt … (A man rides)
b. Ein Mann fährt mit einem Motorrad … (A man rides on his motorbike)
c. Ein Mann fährt mit einem Motorrad nach Völklingen. (A man rides on his motorbike to Völklingen)

(17) a. Dort … (There)
b. Dort kauft er … (There he buys)
c. Dort kauft er einer Frau … (There he buys for a woman)
d. Dort kauft er einer Frau ein Kochbuch. (There he buys a cookbook for a woman)

In contrast to IPG and IPF, POPEL assumes a bi-directional flow of information between the main processing components. At several points during the production of these sentences, feedback information is passed from one level to the previous one. During the production of (16a), for example, the Formulator initially receives only the concept DRIVE, which is realized as a verb. Because the Formulator cannot express this verb without a subject, it passes a request upward which is reformulated so that the Conceptualizer knows which concept to provide. Next, the concept PERS-1 is provided by the Conceptualizer but found unsatisfactory by the Formulator because the it needs a more specific reference to find an appropriate word. Such a reference is found via the *name* role, yielding the concept MAN, which is passed down the cascade. The restrictions at the intermediate knowledge levels, both conceptual and syntactic, are responsible for the rest of the story. They are not discussed here.

Through incremental processing, discourse context has a substantial effect on word order and lexical choice. In (17a), for example, the anaphoric adverb *dort* (there) is chosen rather than a full NP to refer to a recently focused concept, and it comes early in the utterance. The influence of discourse on the form of the utterance is even more evident from the examples (18) and (19), which are produced by POPEL right after (16) and (17) and with the same intentional goals. The changed context is taken into consideration, which leads to adaptations in lexical choice and word order in the sentences and even the order of the sentences themselves. Thus, the organization of content to be expressed is determined by what is topic, implemented by time differences rather than by standard schemas.

(18) Das kauft er ihr in Völklingen.

(19) Dorthin fährt er mit dem Motorrad.

To summarize, POPEL both selects and formulates the concepts that currently get the highest attention before other ones. The surface realisation is adapted to the decisions made by the selection processes. POPEL's use of time spans between conceptual inputs that are incrementally passed on to the Formulator is essentially the same as that in IPG and IPF. A major difference is the provision for feedback in the form of requests from the Formulator to the Conceptualizer. This feedback implies that the Formulator is not passively waiting for input from the Conceptualizer, but takes an active role in requesting concepts necessary for a suitable utterance.

The work on POPEL is currently being followed by work on a more ambitious model called TAG-GEN (Harbusch, Finkler & Schauder, 1991; Finkler & Schauder, 1992). First, this project aims at investigating how incremental input affects further components of the language production process after grammatical encoding. Second, it is based on a different grammar formalism, Tree Adjoining Grammar (TAG; Joshi, 1987). Third, a prototype of this model has been implemented as a parallel system running on a network of computers. And finally, the model is able to realize a limited number of self-repairs when needed by incremental input, for example, *Peter plays … Peter and Mary play ball,* when Mary is added as a second agent at a later time.

11.4 A connectionist model: FIG

The Flexible Incremental Generator (FIG; Ward, 1992) is a connectionist model, somewhat similar to the interactive activation model proposed by Stemberger (1985), and also similar in some respects to the connectionist model by Kalita and Shastri (1987).

FIG strongly deviates from the preceding models due to its connectionist architecture. Perhaps the most striking difference is that it does not explicitly construct syntactic structures at any point during sentence production. Incremental sentence production is viewed as the selection of consecutive words to be uttered, one at a time. Syntactic, lexical and other knowledge mediate the choice of appropriate words at appropriate times, but there is no global structure representing the current syntactic state of affairs.

The linguistic and other knowledge in FIG consists of an associative network with nodes representing conceptual, lexical, and syntactic knowledge. The input to FIG is not substantially different from that in the other models: it is formed by a graph of nodes representing conceptual *notions* (with labels ending on *n*) linked together by thematic *roles* (with labels ending on *r*). As an example, Figure 11.4 depicts the conceptual nodes necessary for sentence (20). However, in contrast to the previous models, FIG feeds all information in the network as a whole into the Formulator. In addition, topicalization is not modelled by timing differences, but by an explicit marker $topic_n$ in the input.

(20) The old woman went to the stream to wash clothes.

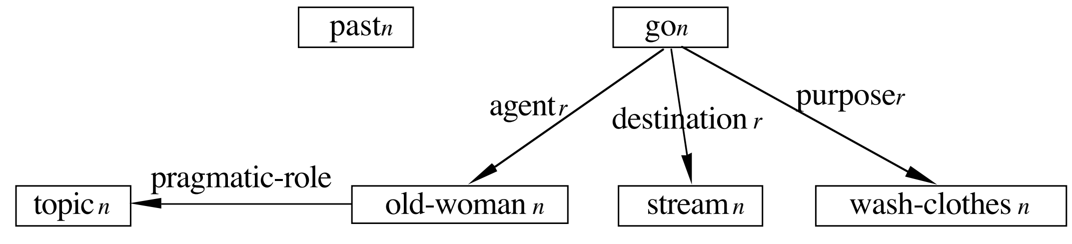


Figure 11.4 An input to FIG.

The computational mechanism responsible for the production of a sentence is based on spreading activation (see Chapter 3). Initially each conceptual node is given a certain amount of activation. The activation then starts spreading through the system, eventually reaching syntactic nodes (called *constructions*) and words. At appropriate intervals, the word with the highest current activation is selected to be uttered as the next word of the sentence. This activation and selection process is repeated until every concept in the input has been expressed. In this way, sentence production proceeds word by word from left to right.

Figure 11.5 illustrates some paths of activation from concepts, mediated by other conceptual and syntactic nodes, to the first word to be uttered, which will be the article *the*. The nodes include words (with labels ending on *w*) and syntactic constructions (with labels ending on *c*) representing phrases and clauses.

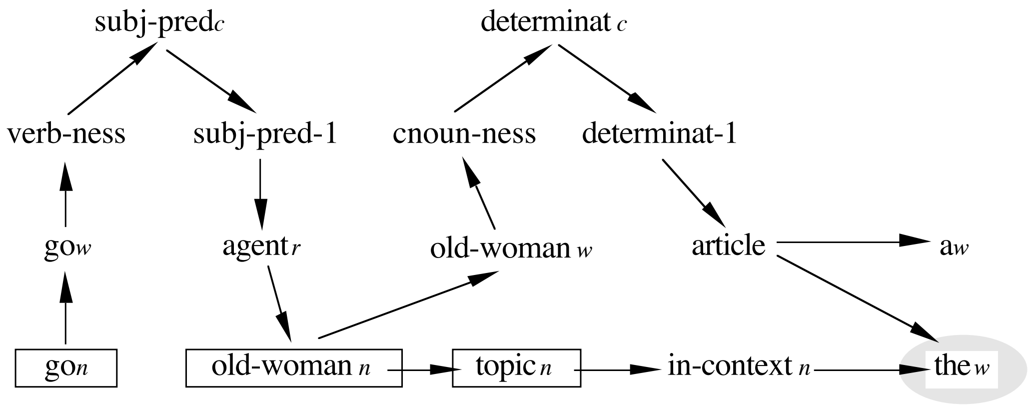


Figure 11.5 Selected paths of activation just before output of the.

In FIG, the activation level of a word represents its current relevance at each time. Therefore, words which are syntactically and conceptually appropriate must be strongly activated at the right moment. In FIG, the representation of the current syntactic state is distributed across the nodes representing phrases and clauses. As an example, a piece of the network in Figure 11.5 is given in somewhat more detail in Figure 11.6, with numerical weights to be multiplied by the activation values flowing on the links. In this network, the constituents of the node $determinat_c$ are encoded in their positional order as *determinat-1* and *determinat-2*.

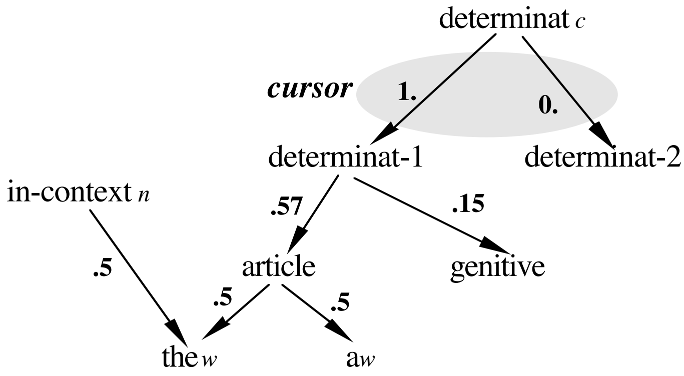


Figure 11.6 Use of the cursor in a syntactic construction.

The current relevance of each constituent in a phrase or clause is determined by the use of *cursors* that move from one constituent to the next. The cursor acts as a filter so that only the currently relevant constituent receives activation. In the network shown in Figure 11.6, for example, the cursor initially points at the first constituent, which is linked to the article and genitive nodes. The cursor puts a weight of 1. on activation flowing to the current constituents, so that activation passes freely, but puts a weight of 0. on the other constituents, which effectively cuts off the flow of activation. After either an article or the genitive has been emitted, the cursor moves to the second node, and so on. Optional constituents will be skipped when there is no concept to activate them. More specifically, the next constituent which receives conceptual activation will become the most highly activated and will be emitted.

Summing up, FIG operates incrementally in the sense that it produces words in time one by one in a strictly left-to-right order. Computation in FIG is guided by the sequencing of its output rather than that of its input. FIG uses local conceptual and syntactic activation patterns to consider many words in parallel and select the appropriate word at the right time. Even if this model does not explicitly construct a syntactic structure, the operations of the cursors implicitly follow the constraints of a distributed grammar that underlies this model.

11.5 Evaluation of the models

It is difficult to compare the models on their actual results, because none of them make predictions that are specific and quantified enough to be tested empirically. Rather, the models illustrate that incremental production itself, as a processing mode, is not only computationally possible but also computationally advantageous, which could be interpreted as an indication of psychological plausibility. This section will mainly compare the theoretical assumptions and computational architectures of the different approaches.

11.5.1 Incremental production and computational parallelism

All models link incremental production to a form of computational parallelism, but in different ways. The AI-based models try to realize every part of the utterance as soon as the corresponding conceptual input becomes available. Each successive conceptual input triggers some grammatical encoding activity in the Formulator, aimed the production of a piece of the utterance, – although sometimes a piece must be held back due to constraints of the grammar and lexicon. Parallelism is dependent on the chunks of conceptual input that are being processed at the same time. In contrast, FIG is driven by the appropriateness of words at each given point in the utterance, where the available conceptual input, the words in the lexicon and the constraints of the grammar all act simultaneously.

Consequently, the AI-based models show how sentence production can already get under way with an incomplete conceptual input, and how variations in the timing of this input are linked to variations in the shape of the resulting utterance, including fine grained effects on word order, as hypothesized by Pechmann (1989; see 11.2.2). In addition, IPG implements the assumption that early constituents take preferential grammatical roles (Bock & Warren, 1985; see 11.2.2). In contrast, FIG uses explicit, separate markers in the input, for example $topic_n$.

Even so, it is not unthinkable that the two views of incrementality could be combined. Ward (personal communication) claims that providing FIG with all necessary concepts for a whole sentence is not strictly necessary and that the model could be adapted to process conceptual input in a piecemeal way. This might perhaps be achieved by activating the nodes of the conceptual input each at different moments in time. How this would interfere with the existing syntactic choice mechanism has not yet been investigated.

A final remark on this matter is that the models presented here hardly take into account how pragmatic factors interact with incremental production. It is not shown, for example, how the demands of incremental production are compatible with the need to manipulate the hearer's attention (McCoy & Cheng, 1991). Only POPEL contains some cohesive devices. The other models assume that any pragmatic considerations will be reflected either in the timing of input to the Formulator or by means of special markers. For example, a focused element in the preverbal message may be given prominence in IPF by passing it to the Formulator earlier than other elements. However, it is never systematically shown that this is adequate for handling pragmatic phenomena beyond single sentences.

11.5.2 Other psycholinguistic phenomena and assumptions

None of the models account very well for the other word order phenomena such as variations due to weight and persistence (see 11.2.2). The IPF model could perhaps be adapted to take into account some measure of processing load. Because grammatical encoding is distributed among separate competing processes that each have their own running time, the complexity of grammatical processing for each constituent could in principle be brought to bear upon the moment when the constituent is integrated in the utterance, and thereby be given a left-to-right position. However, as mentioned before, it seems difficult to reconcile this scheme with incremental processing, because an incremental Formulator prefers to integrate a constituent in the current syntactic structure before its surface features have been completely determined.

To model syntactic persistence, FIG seems to offer the possibility of preactivating 'persistent' syntactic constructions before conceptual input is given.

Similarly, the FIG model accounts for speech errors when, due to contextual or other influences, the activation of an unintended element exceeds that of an intended one, in a way similar to that proposed by Dell (1986; see also Chapter 13). However, it is unclear whether such errors would be similar to the kind of errors in human speech. Dell uses a system of categorized slots and other mechanisms to ensure that errors meet certain constraints; FIG does not seem to have such stringent provisions.

None of the models accounts for the characteristics of SVA errors. With respect to these errors, Vigliocco, Butterworth and Semenza (in press) criticize modular architectures, in particular IPG, because in these architectures, agreement is computed at a separate syntactic level, before and independently of the retrieval of the lexeme of the head noun in the subject NP. This is at odds with cross-linguistic studies indicating that the morpho-phonological marking of the subject head noun affect the rate of SVA errors. Also, some even more marked effects remain altogether unexplained, in particular, the assymmetry between singular and plural errors, and a semantic effect involving distributivity and collective nouns.

The AI-based models and FIG differ in their assumptions with respect to cognitive constraints. FIG represents the syntactic state of affairs locally, whereas the other models keep a large structure in memory for an extended length of time. The latter has been exploited in a model of agrammatism, which, according to Kolk (1987), could result from the speaker's adaptation to some kind of memory impairment. Kolk simulates agrammatism by using a version of IPG implemented with production rules that have a limited time span, so that parts of large structures decay over time. Admittedly, there are other ways to conserve memory in hierarchical models, for example by narrowing the scope of planning at each level (narrowing to the clause during the assignments of functions, or to the phrase during the assignment of positions), but this has not been implemented.

11.5.3 Modularity and interaction

In all models, parallelism is a direct architectural basis for incremental production. Computational simultaneity is deemed necessary for explaining word exchanges (Garrett, 1980). In addition to being psychologically plausible, a parallel mode of processing is more efficient than a serial mode if each processing module is immediately triggered by any fragment of characteristic input. In the AI-based models, there is parallel operation of modules (intercomponent parallelism) as well as parallel operation on different inputs (intracomponent parallelism), with the number of simultaneous processes of the order of the number of conceptual inputs. This kind of parallelism is more efficient than a serial mode in so far as the processing of several inputs overlap. FIG presupposes an interactive rather than a modular architecture, with a more fine-grained, massive parallelism, where each node in the network, either conceptual, syntactic or lexical, spreads activation simultaneously. This kind of parallelism promotes efficiency by simply collapsing the various processing stages.

The information flow in the modular language production models generally flows from the conceptual to the lexical and syntactic levels. Reithinger's POPEL makes use of feedback between the Formulator and the Conceptualizer. Levelt's (1989) arguments against such feedback are supported by (heavily discussed) empirical and simulation results (Levelt et al., 1991). Like Reithinger, Rubinoff's (1992) generator IGEN implements a bi-directional flow of information between the Conceptualizer (which he calls the *planner*) and the Formulator (the *linguistic* component). But in contrast, the feedback in IGEN mainly pertains to word choice rather than syntactic needs. For example, a general concept *rain* can be expressed by various words, including *drizzle* and *pour*, but the Conceptualizer may not know in advance which details of the meaning it should provide in order to allow a suitable choice. The Formulator therefore provides the Conceptualizer with annotations about possible words, some of which may satisfy secondary plans at the conceptual level more than other ones. Thus lexical choice arises as an interplay between the two components.

Within the Formulator, the interaction between lemma retrieval and syntax that the various models assume does not seem to be limited to phenomena such as passivization. Nogier and Zock (1992) view lemma retrieval as a matching operation where both semantics and syntax are highly relevant. Their proposed interaction between lexical choice and syntax suggests that even if grammatical encoding is largely conceptually and lexically driven, lexical choice is also partly syntax driven.

To conclude and sum up the discussion in this chapter, there are a number of computer models of grammatical encoding that are sophisticated from a computational point of view. They have been useful to visualize and test some theories of incremental syntactic processing. However, they have not produced any easily quantifiable results. Up to now, computer models have mainly aided in globally showing that some ways of organising a grammar and syntactic processes are more effective and may be psychologically more plausible than other ones.

Notes

1. The speech errors in this chapter were recorded during spontaneous speech.
2. Overviews of seminal work from these traditions can be found in Kempen (1989) and McDonald (1992). For proceedings of workshops on computer models for natural language production in general, we refer to volumes edited by Kempen (1987b), Zock and Sabah (1988), Dale, Mellish and Zock (1990), Paris, Swartout and Mann (1991), Dale, Hovy, Rösner and Stock (1992), and Horacek and Zock (1993). Much of this work uses the term *generation* rather than production.

11.6 References

Abb, B., Herweg, M., & Lebeth, K. (1993). The incremental generation of passive sentences. *Proceedings of the Sixth Conference of the European Chapter of the ACL* (pp. 3–11). Morristown: Association for Computational Linguistics.

Bock, J.K. (1982). Toward a cognitive psychology of syntax: Information processing contributions to sentence formulation. *Psychological Review, 89,* 1–47.

Bock, J.K. (1986). Syntactic persistence in language production. *Cognitive Psychology, 18,* 355–387.

Bock, J.K. (in press, a). Sentence production: From mind to mouth. In J. L. Miller & P. D. Eimas (Eds.), *Handbook of perception and cognition. Vol. 11: Speech, language and communication*. Orlando, FL: Academic Press.

Bock, J.K. (in press, b). Producing agreement. *Current Directions in Psychological Science*.

Bock, J.K., & Cutting, J.C. (1992). Regulating mental energy: Performance units in language production. *Journal of Memory and Language, 31,* 99–127.

Bock, J.K., & Levelt, W.J.M. (1994). Language production: Grammatical encoding. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 945–984). Orlando, FL: Academic Press.

Bock, J.K., & Miller, C.A. (1991). Broken agreement. *Cognitive Psychology, 23,* 45–93.

Bock, J.K., & Warren, R. (1985). Conceptual accessibility and syntactic structure in sentence formulation. *Cognition 21,* 47–67.

Boomer, D. (1965). Hesitation and grammatical encoding. *Language and Speech 8,* 148–158.

Brotherton, P. L. (1979). Speaking and not speaking: processes for translating ideas into speech. In Siegman, A. W. & Feldstein, S. (Eds.) *Of speech and time: temporal speech patterns in interpersonal contexts*. Hillsdale, NJ: Lawrence Erlbaum.

Dale, R., Hovy, E., Rösner, D., & Stock, O. (Eds.). (1992). *Aspects of automated natural language generation*. Berlin: Springer.

Dale, R., Mellish, C., & Zock, M. (Eds.). (1990). *Current research in natural language generation*. London: Academic Press.

De Haan, P., & Van Hout, R. (1986). Statistics and corpus analysis: A loglinear analysis of syntactic constraints on postmodifying clauses. In J. Aarts & W. Meijs (Eds.), *Corpus Linguistics II: New studies in the analysis and exploitation of computer corpora* (pp. 79–97). Amsterdam: Rodopi.

Dell, G. S. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review, 93,* 283–321.

De Smedt, K. (1990). IPF: An incremental parallel formulator. In R. Dale, C. Mellish & M. Zock (Eds.), *Current research in natural language generation* (pp. 167–192). London: Academic Press.

De Smedt, K. (1994). Parallelism in incremental sentence generation. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 421–447). Norwood, NJ: Ablex.

De Smedt, K. & Kempen, G. (1987). Incremental sentence production, self-correction and co-ordination. In G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology and linguistics* (pp. 365–376). Dordrecht: Nijhoff (Kluwer Academic Publishers).

De Smedt, K., & Kempen, G. (1991). Segment Grammar: A formalism for incremental sentence generation. In C. L. Paris, W. R. Swartout & W. C. Mann (Eds.), *Natural language generation in artificial intelligence and computational linguistics* (pp. 329–349). Boston: Kluwer Academic Publishers.

De Smedt, K., & Kempen, G. (in press). Discontinuous constituency in Segment Grammar. In A. Van Horck. & H. Bunt (Eds.), *Discontinuous constituency*. Berlin: Mouton / Walter de Gruyter.

Finkler, W., & Neumann, G. (1989). POPEL-HOW: A distributed parallel model for incremental natural language production with feedback. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence, Detroit* (pp. 1518–1523).

Finkler, W., & Schauder, A. (1992). Effects of incremental output on incremental natural language generation. *Proceedings of the 10th European Conference on Artificial Intelligence, Vienna* (pp. 505–507). Chichester: Wiley.

Fisher, C., Gleitman, H. & Gleitman, L. (1991). On the semantic content of subcategorization frames. *Cognitive Psychology, 23,* 331–392.

Garrett, M.F. (1975). The analysis of sentence production. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 9, 133–177). New York: Academic Press.

Garrett, M.F. (1980). Levels of processing in sentence production. In B. Butterworth (Ed.), *Language production: Vol. 1, Speech and talk* (pp. 177–220). London: Academic Press.

Goldman-Eisler, F. (1968). Psycholinguisics: Experiments in spontaneous speech. New York: Academic Press.

Harbusch, K., Finkler, W., & Schauder, A. (1991). Incremental syntax generation with tree adjoining grammars. In W. Brauer & D. Hernández (Eds.), *4th International GI Congress on Knowledge Based Systems, Munich* (pp. 363–374). Berlin: Springer.

Hawkins, J.A. (1990). A parsing theory of word order universals. *Linguistic Inquiry, 21,* 223–261.

Hawkins, J.A. (in press). *A performance theory of order and constituency*. Cambridge: Cambridge University Press.

Horacek, H,. & Zock, M. (Eds.). (1993). *New concepts in natural language generation.* London: Pinter.

Joshi, A. (1987). The relevance of tree adjoining grammar to generation. In G. Kempen (Ed.), *Natural language generation: new results in Artificial Intelligence, psychology and linguistics* (pp. 233–252). Dordrecht: Nijhoff (Kluwer Academic Publishers).

Kalita, J., & Shastri, L. (1987). Generation of simple sentences in English using the connectionist model of computation. *Proceedings of the 9th Annual Conference of the Cognitive Science Society, Seattle* (pp. 555–565). Hove: Lawrence Erlbaum.

Kay, M. (1979). Functional grammar. *Proceedings of the 5th Annual Meeting of the Berkeley Linguistic Society* (pp. 142–158).

Kay, M. (1984). Functional unification grammar: A formalism for machine translation. In *Proceedings of COLING84, Stanford* (pp. 75–78). Morristown: Association for Computational Linguistics.

Kelly, M. H., Bock, J. K., & Keil, F. C. (1986). Prototypicality in a linguistic context: Effects on sentence structure. *Journal of Memory and Language, 25,* 59–74.

Kempen, G. (1977). Conceptualizing and formulating in sentence production. In S. Rosenberg (Ed.), *Sentence production: Developments in research and theory* (pp. 259–274). Hillsdale, NJ: Erlbaum.

Kempen, G. (1987a). A framework for incremental syntactic tree formation. *Proceedings of the 10th IJCAI, Milan* (pp. 655–660). Los Altos: Morgan Kaufmann.

Kempen, G. (Ed.), (1987b). *Natural language generation: New results in artificial intelligence, psychology and linguistics*. Dordrecht: Nijhoff (Kluwer Academic Publishers).

Kempen, G., & Hoenkamp, E. (1987). An incremental procedural grammar for sentence production. *Cognitive Science, 11,* 201–258.

Kempen, G., & Huijbers, (1983). The lexicalization process in sentence production and naming: Indirect election of words. *Cognition, 14,* 185–209.

Kempen, G. (1989). Language generation systems. In I. S. Bátori, W. Lenders & W. Putschke (Eds.), *Computational Linguistics: An international handbook on computer oriented language research and applications* (pp. 471–480). Berlin: Walter de Gruyter.

Kolk, H. (1987). A theory of grammatical impairment in aphasia. In G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology and linguistics* (pp. 377–391). Dordrecht: Nijhoff (Kluwer Academic Publishers).

Levelt, W.J.M. (1989). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.

Levelt, W.J.M., & Maassen, R. (1981). Lexical search and order of mention in sentence production. In W. Klein & W. Levelt (Eds.), *Crossing the boundaries in linguistics* (pp. 221–252. Dordrecht: Reidel.

Levelt, W.J.M. & Schriefers, H. (1987). Stages of lexical access. In G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology and linguistics* (pp. 395–404). Dordrecht: Nijhoff (Kluwer Academic Publishers)

Levelt, W.J.M., Schriefers, H., Vorberg, D., Meyer, A.S., Pechmann, T., & Havinga, J. (1991). The time course of lexical access in speech production: A study of picture naming. *Psychological Review*, *98,* 122–142.

Lindsley, J.R. (1975). Producing simple utterances: How far ahead do we plan? *Cognitive Psychology, 7,* 1–19.

Lindsley, J.R. (1976). Producing simple utterances: Details of the planning process. *Journal of Psycholinguistic Research, 5,* 331–351.

Marcus, M. (1980). *A theory of syntactic recognition for natural language*. Cambridge, MA: MIT Press.

McDonald, D. (1992). Natural language generation. In S. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence (2nd ed.)* (pp. 983–997). New York: Wiley.

McDonald, J.L., Bock, J.K., & Kelly, M.H. (1993). Word and world order: Semantic, phonological, and metrical determinants of serial position. *Cognitive Psychology, 25,* 188–230.

McCoy, K. F., & Cheng, J. (1991). Focus of attention: Constraining what can be said next. In C. L. Paris, W. R. Swartout, & W. C. Mann (Eds.), *Natural language generation in artificial intelligence and computational linguistics*. (pp. 103–124). Boston: Kluwer Academic Publishers.

Neumann, G., & Finkler, W. (1990). A head-driven approach to incremental and parallel generation of syntactic structures. In *Proceedings of the 13th International Conference on Computational Linguistics, Helsinki, Vol. 2* (pp. 288–293).

Nogier, J.-F., & Zock, M. (1992). Lexical choice as pattern matching. *Knowledge Based Systems, 5,* 200–212. (Also in T. Nagle, J. Nagle, L. Gerholz & P. Eklund (Eds.) *Current directions in conceptual structures research* (pp. 413–435). Berlin: Springer.)

Paris, C. L., Swartout, W. R., & Mann, W. C. (Eds.). (1991). *Natural language generation in artificial intelligence and computational linguistics*. Boston: Kluwer Academic Publishers.

Pechmann, Th. (1989). Incremental speech production and referential overspecification. *Linguistics, 27,* 89–110.

Pechmann, Th. (1994). *Sprachproduktion: Zur generierung komplexer Nominalphrasen*. Opladen: Westdeutscher Verlag.

Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1972). *A grammar of contemporary English*. London: Longman.

Reithinger, N. (1991). POPEL—A parallel and incremental natural language generation system. In C. L. Paris, W. R. Swartout & W. C. Mann (Eds.), *Natural language generation in artificial intelligence and computational linguistics* (pp. 179–199). Boston: Kluwer Academic Publishers.

Reithinger, N. (1992). The performance of an incremental generation component in multi-modal dialog contributions. In R. Dale, E. Hovy, D. Rösner, & O. Stock (Eds.), *Aspects of automated natural language generation* (pp.263–276). Berlin: Springer.

Ross, J. R. (1967). *Constraints on variables in Syntax*. Doctoral dissertation, Massachusetts Institute of Technology (available from Indiana University Linguistics Club).

Rubinoff, R. (1992). Integrating text planning and linguistic choice by annotating linguistic structures. In R. Dale, E. Hovy, D. Rösner & O. Stock (Eds.), *Aspects of automated natural language generation* (pp. 45–56). Berlin: Springer.

Schriefers, H., & Pechmann, Th. (1988). Incremental production of referential noun phrases by human speakers. In M. Zock & G. Sabah (Eds.), *Advances in natural language generation, Vol. 1* (pp. 172–179). London: Pinter Publishers.

Stallings, L., MacDonald, M.C., & O'Seaghdha, P.G. (1994). What's heavy in heavy-NP shift? *Abstracts of the 7th Annual CUNY Conference on Human Sentence Processing* (p. 32).

Stemberger, J.P. (1985). An interactive activation model of language production. In A. W. Ellis (Ed.), *Progress in the psychology of language, Vol. 1* (pp. 143–186). London: Lawrence Erlbaum.

Vigliocco, G., Butterworth, B., & Semenza, C. (in press). Constructing subject-verb agreement in speech: The role of semantic and morphological factors. *Journal of Memory and Language*.

Ward, N. (1992). A parallel approach to syntax for generation. *Artificial Intelligence, 57,* 183–225.

Zock, M., & Sabah, G. (Eds.). (1988). *Advances in natural language generation: An interdisciplinary perspective*. London: Pinter.