

## The Nature of Constraint-Based Grammar

Carl Pollard<sup>1)</sup>

### O. Introduction

I want to start by thanking the organizers, and especially Prof. Byung-Soo Park, for giving me this opportunity to return to Kyung Hee University after seven long years. After all, it was at Kyung Hee, on the Kwangneung campus, that Ivan Sag and I first publicly presented the so-called standard version of head-driven phrase structure grammar, in a set of forum lectures at the International Conference on Linguistic Studies in August 1989. So it seems especially appropriate that Prof. Park asked me to talk to you here about recent developments in constraint-based grammar.

But I have to admit that when I started to think about what to say, I felt a little overwhelmed. Work in this area has proceeded on so many fronts in recent times -- say since 1989 -- that there is no way to even provide a reasonable summary in just two hours. Looking over the published version of our 1989 lectures, I was a little shocked to realize how much has changed since then, even the basic terminology.

In fact, it dawned on me that back then, the term CONSTRAINT-BASED grammar was not used yet -- instead one spoke of INFORMATION-BASED, or UNIFICATION-BASED, grammar. I think the new term CONSTRAINT-BASED grammar is actually a much more accurate name. After all, the use of the term INFORMATION-BASED in a grammar context really reflects a very special use of that term that was current in the San Francisco bay area during the 1980s, a use which evokes on the one hand the contentfulness of states of affairs in situation semantics compared with the lack of content associated with truth-conditional or possible-worlds semantics, and on the other hand a certain

formal analogy between situation-semantical states of affairs and feature structures. But this terminology never really gained much currency beyond the Center for the Study of Language and Information, and people somehow connected with CSLI.

---

1) This feature article was presented for the 11th Pacific Asian Conference on Language, Information and Computation held at Kyung Hee University on December 20 - 22, 1996.

The other term I mentioned, UNIFICATION-BASED grammar, was much more widely used. But it was never really quite appropriate for the trend in theoretical and computational linguistics that it was intended to denote. After all, UNIFICATION refers to a certain binary algebraic operation on logical expressions or on feature structures conceived of as bearers of partial information, an operation which merges their information content; if we think in terms of feature logic instead of feature structures, this operation essentially corresponds to logical conjunction. Alternatively, the term UNIFICATION also denotes any algorithm that computes this operation.

But it is now widely recognized that we must make a sharp distinction between the formal objects actually licensed by a grammar -- structures (for example the feature structures employed in theories like HPSG for modelling linguistic expressions) -- and feature descriptions, which are used to impose constraints on these structures. In this setting, the grammar is nothing but a set of constraints that structures are required to satisfy in order to be considered well-formed. Of course the unification operation is often used in algorithms that solve the constraints, that is, which find structures that satisfy the grammar; but it is the constraints themselves that are really crucial, not the techniques used to solve them.

If I may make an analogy with mathematical physics: typically a physical theory about some dynamical system -- say a vibrating string, or the solar system -- consists of a set of differential equations. The predictions of the theory are the physical motions that satisfy those equations; or to be more precise, the predictions are certain abstract mathematical objects that satisfy the equations. These abstract objects, called the SOLUTIONS of the equations, are formal models of the actual predicted motions of the system. When we say that the solutions satisfy the equations, the sense of the word SATISFY is exactly the same as the use in logic when we say that a certain structure satisfies a first-order theory. In fact, if we actually formalized differential equations, say within the first-order language of set theory, the solutions would actually be first-order models of the physical theory in the technical logical sense.

The point of all this is that the physical theory consists of the constraints themselves (the equations), and the predictions are the things that satisfy the constraints. The techniques used for solving the equations -- assuming this is possible -- may be of interest to an engineer or a celestial navigator, but they are really only of secondary interest to the theoretical physicist. In exactly the same way, to a theoretical linguist, it is really the constraints themselves -- the

grammar -- that are important, because the solutions of the grammar are the well-formed linguistic objects. Of course the methods for solving the grammar, such as unification, are important to linguistic software engineers, but they are only of secondary interest for theoretical linguistics. This general way of looking at things is summarized in table (1):

(1) Basic notions of Constraint-Based Grammar (CBG)

DOMAIN SOLUTION	CONSTRAINTS	TYPICAL		
		SOLUTIONS	CANDIDATE SOLUTIONS	ACTUAL METHOD
constraint based grammar	the grammar (set of constraints)	structures	well-formed structures	unification; constraint solving; stochastic methods
physical analog	physical theory (the equations)	certain continuous functions	solutions of equations	numerical approximation
logical analog	logical theory (set of wffs)	(logical) structures	models of the theory	model constructions
Chomskian analog	I-language	sequences of phrase markers	licensed structural representations	?

Notice that in addition to the physical and logical analogs, this table also gives rough analogs from Chomskyan linguistic theory. I will have more to say about this later.

## 1. The Nature of Constraint-Based Grammar

In the remainder of this talk, I want to flesh out a little bit the skeletal view of constraint-based grammar that I just sketched. (In a second talk, entitled "HPSG:

An Overview and Some Work in Progress", I will try to convey something of the flavor of current research in constraint-based grammar by looking at some work within the framework of head-driven phrase structure grammar. Of course a number of our colleagues at this conference will also be presenting some of this work in their own talks.)

One way to get a feel for what constraint-based grammar is to look at some of its exemplars. Of course the exemplar closest to home for me is HPSG and other members of the PSG family of grammar frameworks, such as GPSG, JPSG, and KPSG. Another exemplar familiar to many of you is lexical-functional grammar (LFG). In fact, I believe that the difference between LFG and so-called PSG is no greater than the differences among various theoretical proposals within PSG, or even within HPSG itself. As far as I am concerned, then, the separation between PSG and LFG exists more at a sociological level than at the level of scientific content -- but I am aware that not everyone agrees about this. Yet another example is the so-called REPRESENTATIONAL MODULARITY approach proposed by Ray Jackendoff in an important recent critique of Chomsky's minimalist program. A great many other examples come from computational linguistics, such as Martin Kay's FUG and the PATR-II system of Shieber et al., as well as more recent avatars such as TFS, ALE, TDL, Troll, and so forth.

## (2) Some Exemplars of Constraint-Based Grammar

THEORETICAL: Arc-Pair Grammar, LFG, (G,H,J,K,...)PSG,  
Jackendoff's Representational Modularity, ...

COMPUTATIONAL: FUG, PATR-II, TFS, ALE, TDL, Troll, ...

However, as David Johnson and Shalom Lappin make clear in a another very important new critique of Chomsky's MP (called simply "A Critique of the Minimalist Program"), the historically first exemplar of constraint-based grammar was neither LFG nor GPSG, but rather another framework that originated in the mid-to-late 1970's, namely the Arc-Pair Grammar (APG) of Johnson and Postal. Although APG never gained many followers, it is true that most of the key innovations that distinguish constraint-based grammar from its predecessors and competitors were present in APG.

Let me now turn to a more detailed characterization of what constraint-based grammar amounts to. To this end, I'll try to identify some commonalities across

the many frameworks, systems, and research traditions that are generally considered to lie within the domain of constraint-based grammars. Some of these common properties are immediate consequences of the logical architecture I discussed above; others are just methodological commitments or sociological tendencies. Some of the properties I have in mind are listed in (3):

(3) Characteristics of Constraint-Based Grammar

- A. Generativity
- B. Expressivity
- C. Empirical Adequacy
- D. Psycholinguistic Responsibility
- E. Nondestructiveness
- F. Locality
- G. Parallelism
- H. Radical Nonautonomy

A. **GENERATIVITY.** This term has fallen out of fashion, but practitioners of constraint-based grammar still think it is important for a grammatical theory at minimum to tell us what the well-formed structures are. Of course theories are going to differ on such particulars as how many levels of representation there are, and what sort of information each of the levels contains, but I think it is generally agreed that a good theory must tell at least tell us which representations, or ntuples of representations, or derivations, or whatever, are actually predicted. Otherwise the theory doesn't have any empirical consequences. This criterion of generativity entails a certain precision in formulating the theory. Minimally, this includes at least the following three requirements, which are adapted slightly from the three criteria proposed by Geoff Pullum in his celebrated column in *NATURAL LANGUAGE AND LINGUISTIC THEORY* entitled "Formal linguistics meets the boojum."

(4) Three Criteria of Generativity for Grammatical Theory

- (i) it must be determinate whether a given mathematical object is the

kind of mathematical object that is used in the theory for modelling linguistic entities.

- (ii) it has to be determinate whether a given string of symbols (in some formal logic, or in careful natural language) counts as one of the assertions (constraints) of the grammar.
- (iii) given a grammar  $G$  and a mathematical object  $O$  used as a candidate model of a linguistic object (a structural representation or a derivation), it has to be determinate whether  $O$  satisfies the constraints imposed by  $G$ .

The first of these three criteria means that we have to make explicit exactly what the candidate structures are whose well-formedness or ill-formedness is at stake. For example, in HPSG they are feature structures, a certain kind of labelled directed graph. In Chomsky's Barriers theory as formalized by Ed Stabler, they are sequences of phrase markers.

It is impossible to overestimate the importance of the second criterion, because it means that one must actually be able to tell what the theory is. For example, in LFG, the theory is precisely formulated in a combination of context-free grammar, the quantifier-free theory of equality, and a linear logic called "glue language". HPSG is formulated mostly in feature logic, but there is a scandalous exception to this -- namely lexical rules -- which (alas) I will not have time to discuss here. Stabler's version of Barriers theory is formulated in first-order logic. By the way, notice that this criterion does not require that the theory be expressed in an artificial formal language. It could just as well be in plain English, or plain Korean, as long as it is clear what the theory is asserting.

To get a feel for the significance of the second criterion, try to imagine what Einstein's theory of special relativity would look like reformulated by a linguist who rejected it. This is shown in (5):

(5) a. Einstein's equation:

$$E = mc^2$$

b. Linguist's reformulation:

Energy must be in an appropriate licensing relationship with

the mass and the speed of light.

Try to imagine constructing a theory with real empirical consequences based on (5)b!

Last, consider the third criterion. In a fully formalized theory, what this means technically is that given a grammar and a potential structure, it has to be decidable whether the structure satisfies the grammar. This third criterion is not so obviously reasonable as the first two. After all, the first two criteria are satisfied automatically as long as the theory is adequately formalized. What makes criterion (iii) less than straightforward is the fact that in general, given a formal theory and a mathematical structure -- say a first-order theory and a model-theoretic interpretation, it is generally undecidable whether the structure satisfies the theory. But if this is so, then why should we impose this third criterion?

The reason is this. In linguistic theories, the structures that we are working with are, no matter whether they are trees, graphs, or some combination of trees and graphs, are always finite: that is, as formal mathematical objects, they only have a finite number of points or parts or nodes. This is crucially important because of a well-known fact of logic given in (6):

#### (6) Decidability of Model-Checking

For arbitrary  $n$ , given a finite structure  $S$  (i.e. an interpretation) for an  $n$ -order language, and a finite theory  $T$  (i.e. a set of axioms), it is decidable whether  $S$  is a model of  $T$  (i.e. whether  $S$  satisfies the constraints imposed by  $T$ ).

It follows from this that as long as our linguistic theory (together with any background theory which is presupposed by it, beyond logic itself) contains only a finite number of constraints, and as long as they are stated clearly enough for us to be able to figure out what they really mean, we can always decide in a systematic way whether a given candidate structure -- that is, a putative model of the structure of some linguistic expression -- actually satisfies the grammar or not. So it turns out that, even though the third criterion of generativity would be an absurd demand to impose on science in general, it makes eminent good sense to insist on it in the special case of grammatical theory.

I should mention here in passing that criterion (iii) is in no way intended to imply that natural languages conceived as sets of strings are decidable. This does not follow, and I don't know of any good reason to believe this is true.

Unfortunately, some currently influential approaches to the study of grammar fail to satisfy these criteria. One obvious example of this is Chomsky's minimalist program, where nearly all the key concepts, such as MERGE, FULL INTERPRETATION, and REFERENCE SET, are still in search of a precise definition; I refer you to Johnson and Lappin's critique for detailed discussion. Another example is syntactic optimality theory, where such crucial notions as INPUT, OUTPUT, and the GEN function exist only at an intuitive level. A third example, closer to home for me, is the problematic status of lexical rules in HPSG, which, conveniently enough, I do not have time to discuss.

Of course, the point of the generativity criteria is not to deny the value, in appropriate contexts, of loose speculation based on intuitive, imprecise notions. But we should not make the mistake of dignifying such speculation with the term "theory", or mistake the tentative conclusions we draw from such speculation for real scientific results.

B. EXPRESSIVITY. Here I refer not to the expressivity of language itself, but rather to the language in which the grammatical theory is expressed. There used to be an influential point of view which held that the formalism within which grammatical theory was formulated had to be highly constrained. This in turn was supposed to constrain the set of possible grammars, which in turn was supposed to make language acquisition easier to explain. I'm not sure anybody actually believes this argument anymore, but if anyone does, I recommend that he or she stop believing it. For one thing, I don't know of any reason to believe that a member of a relatively small set of languages should be any easier to learn than a member of a relatively large set of languages. This would only be true if the language acquisition device knew in advance what the set of possible options was, but I know of no reason to assume this. And even if the LAD did have this foreknowledge, it is hard to see how cutting down the number of options would help learning. This would only be true if the set of options were finite. Of course, there was a time when it was claimed by adherents of Chomsky's principles-and-parameters approach that the number of "core" grammars was finite; but in the absence of any definition of what actually distinguishes the core from the periphery, this claim is devoid of empirical content.



Another argument sometimes given for constrained formalisms was that this was one way to impose decidability on the languages generated. But as I already mentioned, there is not really any good reason to believe that human languages QUA stringsets actually are decidable, so this kind of argument does not have much force. In fact, we need our formalism to be able to express undecidable problems, since some linguistic problems are undecidable. For example, consider the basic generation problem of finding the syntactic structures corresponding to a given logical form, which in general are only recursively enumerable (SNOW IS WHITE, IT IS TRUE THAT SNOW IS WHITE, IT IS TRUE THAT IT IS TRUE THAT SNOW IS WHITE, etc.).

Within constraint-based grammar, by contrast, the usual view is that the language in which the theory is expressed should be highly expressive, that is, unconstrained. Thus: HPSG is expressed in a feature constraint logic with classical boolean connectives and definite relations. Arc-pair grammar is expressed in first-order logic. And typical constraint-based computational linguistic systems are expressed in languages like PROLOG, LISP, or special-purpose languages built on top of them. Instead of the formalism imposing constraints on possible grammars, it is the grammars themselves that impose the constraints. In fact, if we move from linguistics to any other branch of science, the whole idea that the formalism should constrain the theory appears quite bizarre. Imagine if physicists believed in this! Then we might witness conversations like this:

(7) If physicists required the formalism to constrain the theory

Editor: Professor Einstein, I'm afraid we can't accept this manuscript of yours on general relativity.

Einstein: Why? Are the equations wrong?

Editor: No, but we noticed that your differential equations are expressed in the first-order language of set theory. This is a totally unconstrained formalism! Why, you could have written down ANY set of differential equations!

Of course, this could never happen, because physicists already know that it is the theory that imposes the constraints, not the language in which the theory is expressed.

(8) Expressivity (of the language in which the theory is formulated)

- a. Use an expressively rich language (first-order logic, feature constraint logic, LISP, PROLOG, English, Korean, ...)
- b. The language does not impose constraints on the theory; it is the theory that imposes the constraints.

C. EMPIRICAL ADEQUACY. This is just a fancy phrase for getting the facts right. As constraint-based grammarians and other scientists realize, we often write down a constraint that captures an empirical generalization, without having any idea why the constraint is true. Then we are pounced upon by some well-meaning colleague who complains that our constraint is totally ad hoc and uninteresting because it doesn't follow from any deep principle. Again, imagine what would happen if physicists acted this way:

(9) If physicists required all constraints to follow from "deep principles"

Editor: Professor Einstein, I'm afraid we can't accept this manuscript of yours on general relativity.

Einstein: Why? Are the equations wrong?

Editor: No, but they are totally ad hoc!

Einstein: Ad hoc, ad schmoc! At least they explain otherwise unexplained data about the advance of the perihelion of Mercury.

Editor: But this is nonexplanatory and therefore uninteresting. You need to show that your equations FOLLOW from deep and independently motivated principles!

What is so ridiculous about this, of course, is that every theory has to have some constraints -- axioms -- that don't follow from anything else. And those axioms, of course, no matter whether they are the Head Feature Principle or the Case Filter, can always be accused of being ad hoc and uninteresting. Alas, there is no one deep principle of the universe from which everything follows, at least not as far as we know. Instead, things have to go in the other order: we have to

try to establish wide-coverage empirical generalizations first, and worry later about whether they follow from something else. In any case, logically speaking there is no such thing as a deep principle in a theory, since it is always possible to produce a new set of axioms with the same entailments. Unfortunately, among many linguists nowadays, it is considered more important to propose sweeping fundamental principles, often so vague as to lack any empirical content, than to come up with a constraint that probably gets the facts right over a fairly broad empirical domain. This tendency must be firmly resisted. Resistance to this tendency can be expressed as the methodological principle (10):

(10) The Methodological Principle of Empirical Adequacy

- a. There are no "deep principles", since any theory can be reaxiomatized. In any case, science can only tell how things are, not why. Therefore:
- b. first write constraints that get the facts right, and worry later about which constraints are axioms and which are theorems.

D. PSYCHOLINGUISTIC RESPONSIBILITY. Like their predecessors in the field of generative grammar, constraint-based grammarians still consider themselves to be engaged in an investigation of human linguistic competence. In other words, we take our theories to be about a form of knowledge that resides in the human mind. However, we don't claim that our theories directly reflect anything about human language processing. To use a computational analogy: a constraint-based grammar is more like a data base or a knowledge representation system than it is like a collection of algorithms. To put it another way, the knowledge that our grammars depict is a resource that the human processing mechanisms consults.

Nevertheless, as we come to understand more about human language processing, it is important that our linguistic theories be capable of interfacing with plausible processing models. We must never forget that human processing tasks such as understanding, speech production, and the making of grammaticality judgments are actually feasible. Thus the system of linguistic knowledge that the grammar encodes must in principle be capable of being consulted by human linguistic processes that actually terminate. Thus, even though a grammar is only a

competence model, we do not want it to be based irreducibly on computations that the language user cannot be expected to carry out. This is what I call the methodological principle of psycholinguistic responsibility:

(11) The Methodological Principle of Psycholinguistic Responsibility

Grammars, in spite of being only competence models, must not be based irreducibly on computations that the language user cannot be expected to carry out.

The remaining four characteristics of constraint-based grammars are closely related to psycholinguistic responsibility.

E. NONDESTRUCTIVENESS. This is a generalization of the property that used to be called monotonicity for unification-based grammars. What it means is that the grammar should not irreducibly make reference to operations that destroy existing linguistic structure.

(12) Nondestructiveness

Grammars should not irreducibly make reference to operations (e.g. MOVE) that destroy existing linguistic structure.

Thus there is no raising, no *wh*-movement, no affix-hopping, no head movement. Similarly, there are no null functional categories whose sole purpose is to carry features that must be checked off by moving something into its checking domain. The reason for this is that it is too hard to build plausible processing models that operate destructively on structures already built up. I think this point is easily grasped by most people who have tried to build a parser based upon a linguistic theory, such as transformational grammar, that uses such operations: usually what happens is that one tries to reformulate the theory in a way that eliminates such operations, for example by using chains instead of movement and parsing *s*-structures directly without ever building *d*-structures at all.

In fact in the past it was often argued by practitioners of transformational grammar that transformations were not really at issue, since one always had the option of reformulating the theory in nontransformational terms. Perhaps this is true of GB theory, though we can't say for sure in the absence of an explicit

formalization. However, it seems evident that Chomsky's minimalist program is irreducibly destructive in this sense: there is no way to reformulate it without the operation MOVE, since economy conditions like Procrastinate and the Smallest Derivation Principle are stated in terms of it. It is very hard to see how such an approach can be reconciled with a reasonable processing model. This is because the branch point in a minimalist derivation has to be reached before the syntax interfaces with the articulatory-perceptual system and the conceptual-intentional system. But psycholinguistic research tells us that language is processed incrementally, with syntactic information being continuously integrated with semantic knowledge, encyclopedic knowledge, and even probabilistic knowledge of frequencies of homophonous words. This point can be appreciated by comparing the garden-path sentences in (13) with the structurally identical non-garden-path sentences in (14) (these are from a recent paper by Spivey-Knowlton et al.):

- (13) a. The horse raced past the barn fell.  
 b. The woman warned the lawyer was misguided.  
 c. The bully pelted the boy with warts.
- (14) a. The landmine buried in the sand exploded.  
 b. The woman thought the lawyer was misguided.  
 c. The woman searched for a priest with compassion.

F. LOCALITY. Here the term LOCALITY is used in contradistinction to GLOBALITY. What this means is that given a candidate structure, the question of whether or not that structure satisfies the grammatical constraints must be determined locally, that is, solely on the basis of the given structure, without reference to other "competing" structures.

- (15) Locality (or, the Prohibition on Transstructural Constraints)

Constraints are local in the sense that whether or not they are satisfied by a candidate structure is determined solely by that structure, without reference to other ("competing") structures.

The effect of this is to rule out transderivational constraints or their nontransformational analog, what we might call "transstructural constraints". Again, the principle motivation for adopting this characteristic is the lack of

plausible processing models that incorporate constraints which require comparing alternative structures.

This characteristic places Chomsky's minimalist program outside the realm of constraint-based grammar, since economy requires that any convergent derivation be compared to all other convergent derivations in its reference set. Since so far there is not even a clear definition of reference set, it is unclear in the extreme how the minimalist program could be interfaced with a processing model; but even if it could be, the remaining obstacles are daunting. Here I quote from Johnson & Lappin, who take as their point of departure the following algorithm:

- (16) Algorithm to test a string  $s$  for grammaticality within Chomsky's minimalist program (after Johnson & Lappin)
1. Construct a numeration from the lexical items in  $s$ .
  2. Compute the reference set  $RS$  of convergent derivations from  $N$  to a well-formed  $\langle LF, PF \rangle$  pair.
  3. Use the economy metric to compute the subset  $OD$  of  $RS$  containing the optimal derivations in  $RS$ .
  4. Check if there is at least one element of  $OD$  whose  $\langle LF, PF \rangle$  pair is such that  $PF$  corresponds to  $S$ .

Johnson & Lappin go on to say this:

One could object to this analysis on the basis that more efficient methods for implementing the MP are possible. So for example, one might be able to design an algorithm for testing grammaticality that identifies the set  $OD$  without computing the full  $RS$  of which  $OD$  is a subset.... The burden of argument lies with the critic of our analysis. It is not sufficient to simply assert the logical possibility of an algorithm for implementing the MP that is more efficient than the one we assume here.... Second, even if a more efficient algorithm can be constructed, any implementation of an economy-of-derivation model will still involve conceptual if not computational complexity beyond that required by a local constraint grammar.

Once again, I refer you to their paper for the detailed arguments.

Likewise, the locality criterion places syntactic optimality theory outside the realm of constraint-based grammar. To see why, let me first remind you of the overall architecture of optimality theory:

(17) Overview of Optimality Theory (Prince and Smolensky 1993)

- a. There are two sets of structures, INPUTS and CANDIDATES.
- b. The function GEN maps each input I to a subset GEN(I) of candidates.
- c. Language-particularity consists of a ranking of a universal set of constraints.
- d. Given a subset S of CANDIDATES and a member C of S, S is OPTIMAL in S provided, if C violates any constraint, then every other member of S violates some higher constraint.
- e. Given an input I, the OUTPUT associated with I, OUTPUT(I), is the set of optimal members of GEN(I).

Now consider the task of testing a string for optimality within optimality theory. This is shown in (18).

(18) Algorithm to test a string s for grammaticality within syntactic Optimality Theory

- a. Find the set of inputs INPUT(s) that correspond to s.
- b. For each member I of INPUTS(s), compute GEN(I).
- c. For each I in INPUTS(s), check whether the set OUTPUT(I) of optimal members of GEN(I) is nonempty.  
If this is the case for some I, the string s is grammatical.

This is a tall order. Even though the question of determining the set of inputs corresponding to a string is one that has not been discussed in syntactic Optimality Theory, let's give OT the benefit of the doubt and suppose this can be done. The real problem is that steps b and c depend crucially on the function GEN being well-defined. Unfortunately, the function GEN in syntactic OT has become notorious precisely because nobody ever defines it. Instead, in a typical OT paper, we are presented with a tableau of candidates, and are supposed to take it on faith that what we are shown actually is the output of GEN for some input, or at least that any members of GEN of that input which are missing from

the tableau are obviously not optimal. To make things worse, usually the sets of potential inputs and potential candidates are not clearly defined either, so that syntactic OT does not even satisfy the first criterion of generativity. Until these undefined notions are made clear, there is no point even talking about whether syntactic OT could be interfaced with a plausible processing model. And even if these undefined notions were defined, syntactic OT would still be subject to the same criticisms that Lappin and Johnson level against the minimalist program.

G. PARALLELISM. It's widely recognized, both within and without constraint-based grammar, that linguistic theory must make reference to different levels of representation. Of course this idea is long familiar from the T-model architecture of the Extended Standard Theory and GB theory, and even the Minimalist Program retains the two interface levels LF and PF. Constraint-based grammars also recognize different levels of representation, although the identity and nature of the levels varies from theory to theory. Some examples are shown in (19)

(19) Some levels of representation in constraint-based grammar

	Syntactic Theory	Logico- Semantic Representation	Grammatical- Relational Structure	Phonetic- Prosodic Structure
HPSG	DAUGHTERS		CONTENT	ARG-STRUC
PHONOLOGY			VALENCE	
LFG	c-structure	s-structure	f-structure, a-structure	prosodic structure
RM	syntactic structure	conceptual structure		phonological structure
GB analog	s-structure	LF	d-structure	PF

The difference between the constraint-based grammars and the T-model, of course, is that none of the levels is derived by transforming one of the others. Instead, the different levels exist in parallel, being mutually constrained by the



grammar. Thus a linguistic expression is represented by an  $n$ -tuple of structures, or alternatively by  $n$  features of a feature structure.

(20) Mutually Constrained Parallelism

No level of representation is derived by transforming (= destructively operating upon) another level. Instead all levels are parallel and mutually constrained by the grammar. Thus a linguistic expression is represented by an  $n$ -tuple of structures, or alternatively by  $n$  features of a feature structure.

This architecture of mutually constrained parallel levels is defended at length in the Jackendoff paper I mentioned above. I can't summarize his arguments here, but let me mention just one of Jackendoff's points. It's been remarked by a number of people over the past 20 years that both the T-model and the MP model are problematic with respect to the issue of lexical insertion. The problem is that if lexical insertion is early, as is usually assumed, then the phonological and semantic information borne by the lexical entries has to be dragged around uselessly through the syntactic derivation, only to be handed off to PF and LF at the branch point of the derivation. As Ivan Sag has put it, it is as if the syntax has to lug around two locked suitcases, one on each shoulder, only to turn them over to other components to be opened. Of course this view of things is completely at odds with the psycholinguistic evidence that language processing consults the various levels of information in a flexible and interleaved fashion. By contrast, in the parallel architecture of constraint-based grammar, there is no lexical insertion. Instead, the lexical entries are just small-scale constrained parallel structures. If we think of the constraints as recursively generating all the well-formed  $n$ -tuples of parallel structures, then we can think of the lexical entries as forming the base of the recursion. This is summarized in (21):

(21) The lexicon in constraint-based grammar

There is no lexical insertion. Instead a lexical entry is just a small scale  $n$ -tuple of constrained parallel structures (or a feature structure with  $n$  features). If the constraints recursively generate the well-formed  $n$ -tuples, then the lexical entries form the base of the recursion.

**H. RADICAL NONAUTONOMY.** The last characteristic of constraint-based grammars that I want to mention is what might be called radical nonautonomy, in contradistinction to traditional assumptions about the autonomy of syntax. This is really just a corollary to parallelism. As we've seen, the grammar consists of assertions that mutually constrain several different levels of structure. Some of these constraints may apply only to one level, say to a phonological level or to a level dealing with grammatical relations. But typically, constraints in constraint-based grammar are interface constraints, in the sense that they mutually constrain two or more levels. Thus we have, e.g., syntax-phonology interface constraints, such as linear precedence theory; or syntax-semantics interface constraints, such as binding theory and constraints on scope of quantifiers and operators; or phonology-pragmatics interface constraints, such as the relation between pitch accent and contrastive focus. The last thing we want is an autonomous theory of syntax. Instead what we need are theories that deal simultaneously with all linguistically relevant factors, be they phonetic, morphological, syntactic, semantic, or pragmatic. And once we get serious about interfacing the theory of competence with processing models, nonlinguistic factors such as world knowledge, frequency considerations, and the beliefs and goals of speakers must also be brought into the picture. It seems to me that, among the existing options, constraint-based grammar has the highest potential to rise to this challenge.

(22) Examples of interface constraints

Syntax-Phonology: linear precedence (LP) constraints

Syntax-Semantics: binding theory; quantifier and operator  
scope

Phonology-Pragmatics: contrastive focus and pitch accent

Argument Structure-Syntax: immediate dominance (ID) rules

Argument Structure-Semantics: linking theory

**Ohio State University.**

**Department of linguistics**