Pergamon

J. Neurolinguistics, Vol. 10, No. 2/3, pp. 231–249, 1997
© 1997 Elsevier Science Ltd
All rights reserved. Printed in Great Britain
0911–6044/97 $17.00+0.00

PII: S0911–6044(97)00007–9

SYNTAX FACIT SALTUM: COMPUTATION AND THE GENOTYPE AND PHENOTYPE OF LANGUAGE

ROBERT C. BERWICK

Center for Biological and Computational Learning, Massachusetts Institute of Technology, U.S.A.

1. INTRODUCTION: LANGUAGE, BIOLOGY AND COMPUTATION

Human language has long captured the imagination of biological researchers. However, the gulf separating 'computation', 'biology', and 'language' has been equally long-standing—in large measure resulting from the gap between linguistic and biological description: we do not expect to literally find a 'passive grammar rule' inside a person's head. Similarly, we evidently do not find a corresponding passive rule-specific 'dysphasia' that destroys one's ability to say, ''This problem was solved by the student'' while sparing, along with the rest of the grammar, the ability to say ''The student solved this problem''.

The puzzle is a classic biological one: how to bridge between a 'genotype' and a 'phenotype', in this case perhaps the most complex behavioral phenotype we know. Linguistics has cast natural languages's intricate and ultimately behavioral 'outer form' or 'phenotype' at an abstract level far removed form language's computational and biological 'inner form' or 'genotype'. Thus, even though linguistic science's successful program the past 40 years has resulted in perhaps the richest description of a human 'genotype to phenotype' mapping that we know of—the initial substrate for human language and how language develops in an individual—until recently, progress at a more 'reductionist' level has been far more halting.

The aim of this article is to show how recent developments in linguistic theory dubbed the ''minimalist program'' (MP) [1, 2] might help bridge the biology–language 'abstraction gap', because the minimalist program shows that despite its apparent surface complexity, language's core might in fact be much simpler than has previously been supposed. For neuroscientists pursuing clues left by the linguistic phenotypes's 'fault lines' down to the level of the real genotype, as exemplified in the work of Gopnik and colleagues [3, 4, 5], or even the so-called 'candidate gene approach', this is a promising development. The minimalist program is eliminative in exactly the right sort of way, since it boils down all syntactic relations and rules, including transformations, to feature matching—thus tying linguistic theory more tightly than ever to accounts such as Gopnik's model of Specific Language Impairment.

More generally, the minimalist program serves as a case study for how a complex behavioral phenotype can emerge from the interactions of a much simpler 'genotype'. In particular, the minimalist program posits that the human 'syntactic engine'consists of just three components: (1) words, that is, word semantics and word morpho-phonology; (2) word features; and (3) a simple operation that glues together or *merges* words and word complexes.

This article demonstrates how just these three components interact to yield many, perhaps all, of the 'special design features' of human syntax. By 'design features' we simply mean familiar properties of human language such as its *digital infinity* or recursive generative capacity; the ability to *displace* or *move* phrases from their natural argument positions, as in, *This student, I want to solve the problem* where the subject of the verb *solve*, namely *the student*, appears at the front of the sentence instead of in its normal position before the verb; *core grammatical relations* such as subject and object; and *locality constraints* that prevent movement from acting over unbounded deomains—in *Who do you wonder Bill thinks solved the problem, who* cannot be interpreted as the subject of *solve.*

However, natural languages deploy only about a half-dozen syntactic relations and predicates out of a potentially infinite set of logically possible relations. For example, human languages often match verbs to objects (in terms of predicate–argument structure); require agreement between tense/inflection and subjects as in the case of subject–verb person/number agreement; or verbs may select either subjects or objects, as in the familiar contrast between *John admires honesty and Honesty admires John.* Yet most logically possible syntactic rules and relations are unattested—for instance, there is no analog of 'object of', say *subject–object–of*, where the subject and object of a sentence must agree.

For the neurobiologist/language scientist, it is important to understand why we see the design we do, in order to further characterize possible deficit patterns, ultimately to link biology to linguistic description. From a logical or communicative standpoint, these particular 'design properties' are otherwise mysterious. For instance, there is no immediately obvious computational or communicative reason why languages ought not to relate subjects and objects. Communicatively, a sentence's subject, usually an 'agent' and its object, usually the 'affected recipient', form just as natural a class as subject and predicate; further, as is easy to see from the transitivity of conditional probabilities, nothing seems to block a computational relation between subject and object. The ultimate explanation must be, obviously, biological, but from the view here, not at the level of 'expressiveness' or 'communicative efficiency'. This article offers an alternative, deeper possibility: why human syntax looks the way it does rather than some other way—why natural languages can have an *object-of* relation but not a *subject–object-of* relation—follows from the fundamental principles of the syntactic engine itself.

Further, the minimalist reformulation has important consequences for models of language processing, and so ultimately descriptions of the linguistic phenotype. The most 'minimal' conception of a processor or parser for natural language takes the relation between basic parsing operations and the abstract linguistic system as simply the identity function. As it turns out, this leads to the most efficient processor possible, and at the same time replicates some of human language's known psychophysical, preferential 'blind spots'. For example, in sentence pairs such as *John said that the cat died yesterday/John said that the cat will die yesterday, yesterday* is (reflexively) taken to modify the second verb, the time of the cat's demise, even though this is semantically incoherent in the second sentence.

If this approach is correct, then linguistics may have reached a better level of primitive description in order to proceed biologically. In this sense, using familiar Darwinian terms, the syntactic system for human language is indeed, like the eye, an "organ of extreme complexity and perfection". However, unlike Linnaeus' and Darwin's slogan shunning the possibility of discontinuous leaps in species and evolution generally—*natura non facit saltum*—we advocate a revised motto: *syntax facit saltum*—syntax makes jumps—in this case, because human language's syntactic phenotype follows from interactions amongst its deeper components to give it a special character all its own.

The remainder of this article is organize as follows. Section 2 serves as a brief 'cook's tour' of the minimalist program as an essentially 'feature driven' syntactic theory. We outline how sentence derivations work in the minimalist program, running through two step-by-step examples, the first simple sentence; the second, a more complex case with movement, paying particular attention to how derivations in the minimalist program connect to the feature-based account of SLI. Section 3 turns to sentence processing and psychophysical 'blindspots'. It outlines a specific parsing model for the minimalist system, based on earlier computational models for processing deterministically, strictly left to right. It then shows how reflexive processing preferences like the one described above can be accounted for.

2. THE MINIMALIST PROGRAM: A COOK'S TOUR

As it is familiar, over the past 40 years linguistic science has steadily moved from less abstract, naturalistic surface descriptions to more abstract, 'deeper' descriptions—rule systems or generative grammars. The minimalist program can be regarded as the logical endpoint of this evolutionary trajectory. While the need to move away from mere sentence lists seems clear, the rules that linguists have proposed have sometimes seemed, at least to some, ever farther removed from biology or behavior than the sentences they were meant to replace. Given our reductionist aim, it is relevant to understand how the minimalist program arose out of historical developments of the field, partly as a drive towards a descriptive level even farther removed from surface behavior. We begin therefore with a brief review of this history.

2.1. Evolution of minimalism: the historical context

It should first be noted the 'abstraction problem' is not unfamiliar to biologists. We might compare the formal computations of generative grammar to Mendel's Laws as understood around 1900—abstract computations whose physical bases were but dimly understood, yet clearly tied to biology. In this context one might do well to recall Beadle and Beadle's comments [6] about Mendel, as noted by Jenkins [7].

> There was no evidence for Mendel's hypothesis other than his computations and his wildly unconventional application of algebra to botany, which made it difficult for his listeners to understand that these computations *were* the evidence.

In fact, as we suggest here, the a-biological and a-computational character sometimes (rightly) attributed to generative grammar resulted not because its rules were abstract, but rather because rules were not abstract enough. Indeed, this very fact was duly noted by the leading psycholinguistic text of that day: Fodor *et al.*'s *Psychology of Language* (1974:368, [8]), which summarized the state of psycholinguistic play up to about 1970:

> ... there exist no suggestions for how a generative grammar might be concretely employed as a sentence recognizer in a psychologically plausible system.

In retrospect, the reason for this dilemma seems clear. In the initial decade or two of investigation in the era of modern generative grammar, linguistic knowledge was formulated as a large set of language-particular, specific rules, such as the rules of English question formation, passive formation, or topicalization. Such rules are still quite close to the external, observable behavior—sentences—they were meant to abstract away from. By 1965, the time of Chomsky's *Aspects of the Theory of Syntax* [9], transformational rules consisted of two parts: *a structural description*, generally corresponding to a surface-orientated pattern description of the conditions under which a particular rule could apply (an 'IF' condition), and a *structural change* marking out how the rule affected the syntactic structure under construction (a 'THEN' action). For example, the 'passive rule' might be formulated as follows, mapping *Sue will eat* *the ice-cream* into *The ice-cream will be+en eat by Sue*, where we have distinguished pattern-matched elements with numbers beneath:

Structural description (IF condition):

Noun phrase	Auxiliary Verb	Main Verb	Noun Phrase
1	2	3	4
Sue	will	eat	the ice-cream

Structural change (THEN):

Noun phrase	Auxillary Verb	be+en	Main Verb	by	Noun Phrase
4	2		3		1
The ice-cream	will	be+en	eat	by	Sue

In the Aspects model a further 'housekeeping' rule would next apply, hopping the *en* affix onto *eat* to form *eaten*. This somewhat belabored passive rule example underscores the non-reductionist flavor of earlier transformational generative grammar: the type and grain size of structural descriptions and changes simply do not mesh well with the biological descriptions of, for example, observable language breakdowns. Disruption does not seem to occur at the level of individual transformational rules, not even as structural descriptions and changes gone awry generally.

Moreover, given such rule diversity and complexity, even by the mid-1960s the quasi-biological problems with surface-oriented rules—problems of learnability and parsability, among others—were well known: how could such particular structural conditions and changes be learned by children, given the evidence that linguists used to induce them was so hard to come by? The lack of rule restrictiveness led to attempts to generalize over rules, for example, to bring under a single umbrella such diverse phenomena as topicalization and question formation, each as instances of a single, more general, 'Move wh-phase'operation. By combining this abstraction with the rule 'Move Noun phrase', by the end of the 1970s linguists had arrived at a replacement for nearly all structural changes or 'displacements', a single movement operation dubbed 'Move-alpha'. On the rule application side, corresponding attempts were made to establish generalizations about constraints on rule application, thereby replacing structural descriptions—for example, that noun phrases could only be displaced to positions where they might have appeared anyway, as in our passive example.

By the mid-1980s, the end result was a system of approximately 25–30 interacting "principles", the so-called "principles and parameters" or "government and binding" approach. Figure 1 sketches its general picture of sentence formation, shaped like an inverted Y. This model engages two additional representational levels to generate sentences: first, *D-structure*, a canonical way to represent predicate–argument thematic relations and basic sentence forms—essentially, 'who did what to whom', as in *the guy ate the ice-cream* where *the guy* is the 'consumer' and *ice-cream* is the item consumed; and second, *S-structure*, essentially a way to represent argument relations after 'displacement'—like the movement of the object to subject position in the former passive rule—has taken place. After the application of 'transformations' (movement), S-structure splits, feeding sound (phonological form, PF) and logical form (LF) representations to yield (sound, meaning) pairs.

Overall then, on the principles-and-parameters view, sentences are *derived* commencing with a canonical thematic representation that conforms to the basic tree structure for a particular language, and then mapped to S-structure via a (possibly empty) sequence of displacement operations. At this point, S-structure splits into two, being read off into PF and LF. For instance,

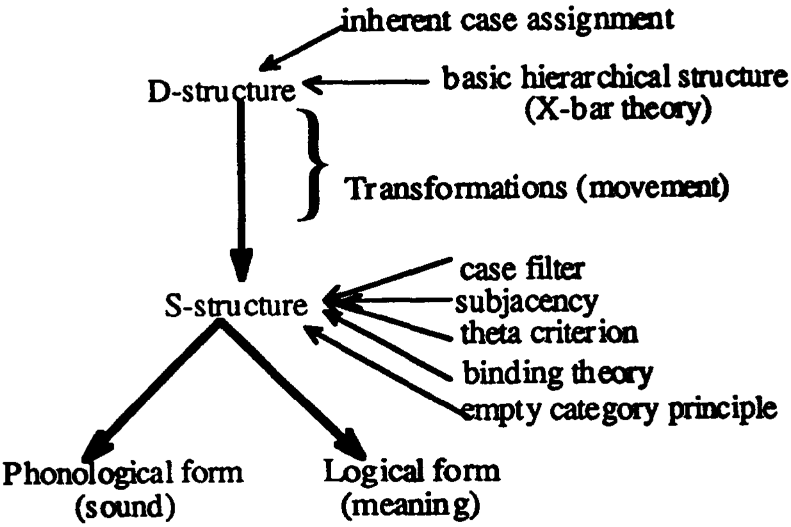

Fig. 1. A conceptual picture of the 'traditional' transformational generative grammar framework (applying equally to the Extended standard theory, government–binding, or principles-and-parameters approaches). Thin arrows denote constraints that possible sentence forms must satisfy, like the case filter. We do not describe all the depicted constraints in this article.

one could start with *the guy ate the ice-cream* in hierarchical form with a thematic or D-structure; via displacement of the *ice-cream* this initial representation can be mapped to the topicalized form, *ice-cream, the guy ate*. Conceptually, in the principles-and-parameters approach, sentence generation can be viewed as starting at D-structure and then 'running a gauntlet' through a set of constraint boxes placed at D-structure and S-structure, as shown in the figure. A sentence is completely well formed if it passes all the constraints and emerges at the two interfaces of phonological form and logical form as one or more sound, meaning pairs.

Akin to atomic theory, this small set of constraints may be recombined in different ways to yield the distinctive syntactic properties of diverse natural languages, just as a handful of elements recombine to yield many different molecular types. For example, one of the principles, X-bar theory, constrains the basic D-structure 'tree shapes' for phases—whether phases appear in function–argument form, as in English verb–object or preposition–object combinations, such as *eat ice-cream* or *with a spoon*, or arguement–function form, as in Japanese object–verb or postposition–object combinations, such as *ice-cream-o tabeta* or *spoon-ni*.

The X-bar 'module' constrains just a small part of the ultimate surface form of sentences and must conspire with other principles to yield the surface complexity that one actually sees. In order to replicate the passive rule, at least three other general principles constraining displacement and S-structure come into play. One such constraint is the so-called *theta criterion*: if one pictures a verb taking some number of arguements—its thematic roles, such as *drink* requiring something to be drunk—then at the end of a derivation, all of the verb's arguments must have been 'discharged' or realized in the sentence, and every possible argument in the sentence must have received some thematic role. A second constraint is the *case filter*: any pronounceable noun phrase, such as *the guy*, must possess a special feature dubbed *Case*, assigned by a verb, preposition or tense/inflection.

Now the former passive rule follows as a 'theorem' from these more basic principles. Starting from the 'D-structure' or the thematic representation *was eaten ice-cream*, since *eaten* does not assign Case (analogously to an adjectival form, like *tired* or *happy*), the object *ice-cream* must move to a position where it does get case—namely, the position of the subject, where *ice-cream* can receive case from the inflected verb *was*. We thus derive the surface form *the ice-cream was eaten*. The thematic association between *eat* and *ice-cream* as the 'material eaten' is retained by a bit of representational machinery: we insert a phonologically empty (unpronounced) element, a *trace*, into the position left behind by *ice-cream* and link it to *ice-cream* as well. In a similar fashion one can show that approximately 30 such constraints suffice to replace much of syntax's formerly rule-based core.

2.2. *The minimalist program*

The minimalist program takes the principles and parameters approach one step further: it aims to eliminate all *representations* and *relations* that can be derived from more primitive notions. Syntax still mediates form and meaning in the classical Saussaurean sense, as in Fig. 1 with its paired sound and meaning ''interfaces''—but the representations of D-structure and S-structure are eliminated. To build syntactic objects and relations, minimalism invokes only the notion of a 'word' construed as a list of features plus a generalizated hierarchical derivation operator, called *Merge*. For example, it is Merge that glues together *eat* and *ice-cream* to form the verb phrase *eat ice-cream* and tacks the *ed* morpheme onto the end of *eat* to form *eaten*; a sequence of Merges generates a sentence. In fact, relationships among syntactic objects established by Merge constitute the totality of syntactic structure, and, as we shall see, also fix the range of syntactic relations. In other words, those elements that enter into the Merge operation are precisely those that can be syntactically related to each other. Merge thus delimits the 'atoms' and 'molecules' visible for 'chemical combination'. At the sound–meaning interfaces, the only available entities are syntactic objects and the syntactic structures these objects form. These entities contain inherent word features that impose constraints on articulatory generation or parsing and conceptual–intentional interpretation. What 'drives' the generative process is feature matching and feature elimination, as we now describe.

2.2.1. Deriving sentences in the minimalist approach. To see how this generative machinery works, let us consider a concrete example that we will follow through the remainder of this article. The following two figures illustrate, with Fig. 2 providing a conceptual overview and Fig. 3 more detail. We retain the basic syntactic categories from previous syntactic models, considered as features, both open class categories such as n(oun) and v(erb), as well as grammatical categories like d(eterminer), t(ense) (or i(nflection)), c(complentizer), and so forth. Conceptually, in Fig. 2 we begin with an unordered 'bag' of words (formally, a multiset, since some words may be repeated), where words are just feature bundles as we describe in more detail later. In our example, we begin with *(the, guy, drank, the, wine)* and via four derivational steps, for Merges, wind up with the syntactic structure corresponding to *the guy drank the wine*, which is then spun off to both phonological and semantic interpretation. We should emphasize at the outset that these figures depict just *one*possible, successful derivational sequence. In fact, with five words there are 51, or 120, possible basic derivational possibilities, but most of these, as we shall see, do not lead to well-formed structures.

Adhering to the chemical analogy of sentence derivation, minimalism deploys Merge to combine words into larger, hierarchical superwords via a notion somewhat like chemical valency. All structure building is feature driven, via words with formal (F), phonological/sound (P), and semantic (S) features. Figure 2 depicts these as F, P and S features in an initially unordered word 'soup' (a lexicon). Roughly, phonological features are those that can be interpreted, or 'read' by the articulatory/perceptual interface—such as classical distinctive features that refer to articulation, like +/-Coronal; semantic features are those that can be read by the conceptual/intentional interface—such as +/-Past; while formal (or syntactic) features include those such as +/-Tns or +/-Case that play no role in sound or meaning. Syntactic features also encompass the traditional notion of *selection*, as in the sense of agreement or a verb–argument relation: a feature attached to a word can select for a particular syntactic category feature to its right or left. Following Stabler [10, 11], we use the notation $=x$ to denote such a requirement; for example, the feature $=n$ means that a verb like *drink* could select a word marked with the feature $=n$ (for noun) to its right or left.

Merge combines two words, a word plus an affix, or two word complexes into a new 'superword', with one of the two elements being selected as the 'head' of the new hierarchical structure, as shown schematically as the combination of two vertical lines into a triangle or two triangles into a larger one. Merge is triggered in two ways: (1) either if *+f* and *-f* formal features on two words or word complexes can cancel, erasing the formal feature *f*; or (2) by $a=x$ feature can select a $+x$ category. For example, we take *the* to be a determiner, *+det*, selecting the feature *n*(oun), so it has the formal features *-det*, $=n$; while *wine* is marked *+n*, *-Case*. The $=n$ feature can select the $+n$ feature, so Merge is possible in this case. (Right/left order is irrelevant for selection; we put to one side the important question of how actual word order is fixed, e.g. why the combination *wine the* is barred.) Merging these two words, *the* is taken as the head of a new hierarchical complex, which one can write as {the {*the wine*}}, and which would traditionally have been written as a phrase structure tree. The process of matching and canceling features, or matching and selecting features, is called *feature checking*. Note that it is possible for Merge to fail, if features do not cancel or match: for instance, we cannot Merge *wine* and *guy*. Finally, it is important to add that Merge is driven by a locality notion of 'economy': a feature *-f* must be checked "as soon as possible"—that is, by the closest possible corresponding *+f* feature.

After a Merge, any features that remain uncanceled are copied or projected to the top of the new hierarchical structure, so our example complex has the features *+det*,v *-Case*; conventionally, a noun phrase. (We could just as easily envision this as 'copying' the entire word *the* to the head of the new structure, as shown in Fig. 2.) Note that from this perspective, it is only words and affixes—the leaves of a syntactic structure—that have features; the head of a hierarchical structure receives its features only via inheritance.

This Merger process repeats itself until no more features can be cancelled, as shown in Fig. 2, and in detail in Fig. 3—note that after step 4, all formal syntactic features have been eliminated and only sound and meaning features remain to be 'read' by the phonological and conceptual/intentional machinery, a process dubbed 'spell-out'. Note that in fact spell-out is possible at any time, so long as the structure shipped off to PF or LF is well formed.

Step by step, generation proceeds as follows (see Fig. 3). Selecting a possible Merge at random, *The*{*+det*, $=n$} can combine with *wine* {*+det*, $=n$}, selecting the $+n$ feature, and yielding a complex with *+det*, *-case* at its root. Note that we could have combined *the* with *guy* as well. For the next Merge, one might combine either *the* with *guy* or *drank* with *the wine*, selecting the *+det* feature and canceling the *-case* requirement corresponding to the noun phrase argument *wine*—this corresponds to a conventional verb phrase. The root of this complex still has two unmet feature requirements: it selects a noun ($=n$), and assigns a case feature (*+case*). Note that an attempted Merge of *drank* with *the* before a Merger with *wine* would be premature: the *v*, = *det*, features would be percolated up, to a new *wine-the* complex. Now *wine* could no longer be combined with *the*. (On the other hand, there is nothing to

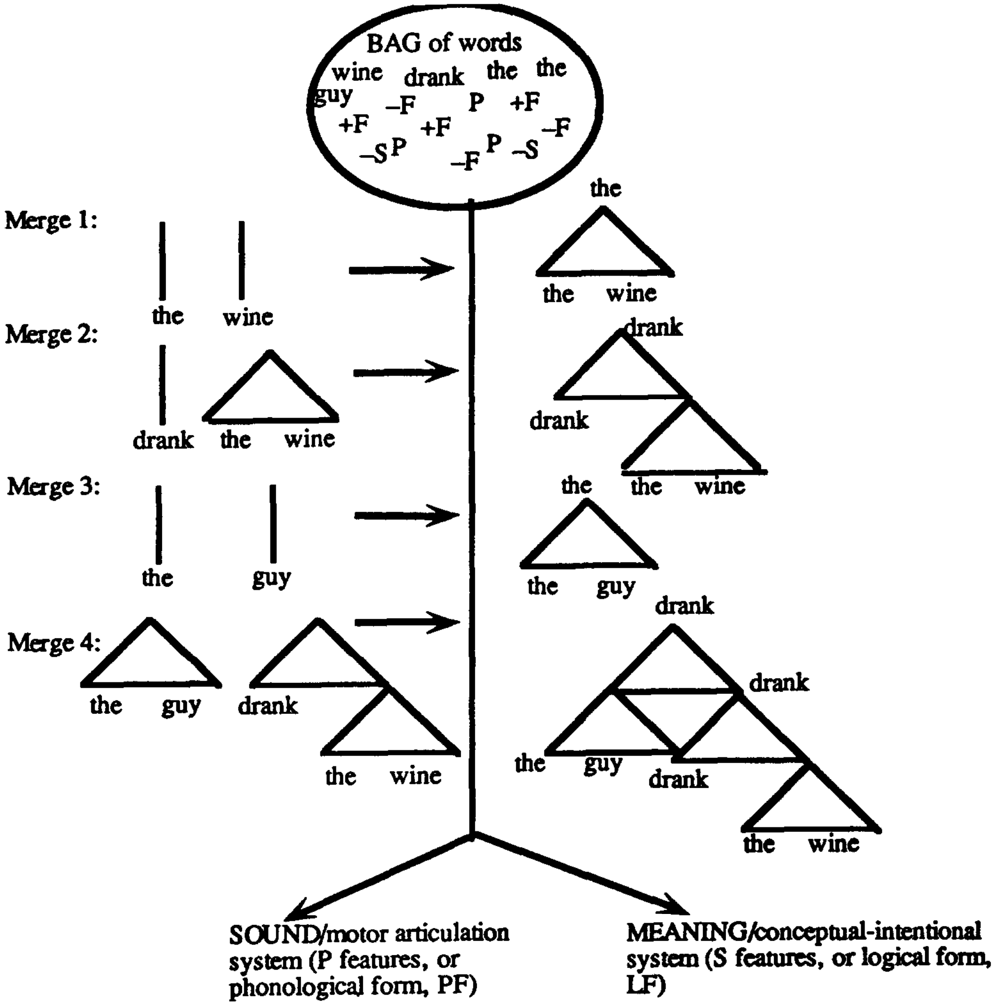

Fig. 2. Merge maps from an initial word sequence to a (sound, meaning) pair—representations containing only phonological or semantic features, respectively. A sequence of Merges constitutes a derivation in the minimalist program, generating a sentence from an initial unordered word set. P, S and F stand for phonological, semantic and formal (syntactic) features, respectively.

syntactically block the sentence form, *the wine drank the guy*; presumably, this anomaly would be detected by the conceptual–intentional interface.)

Proceeding then with the verb phrase path, depending on whether *the* and *guy* had been previously merged, we would either carry out this Merge, or, for the fourth and last step, Merge *the guy* with the 'verb phrase', in the process canceling the *-case* feature associated with *the guy*. At this point, all formal features have been eliminated, save for the *v* feature heading the root of the sentence, corresponding to *drank* (in actual practice this would be further Merged with a tense/infl(ection) category). We can summarize the generation process as follows:

1) Merge 1: combine *the* and *wine*, yielding *the wine.*

1) Merge 2: combine *drank* and *the wine*, yielding *drank the wine.*

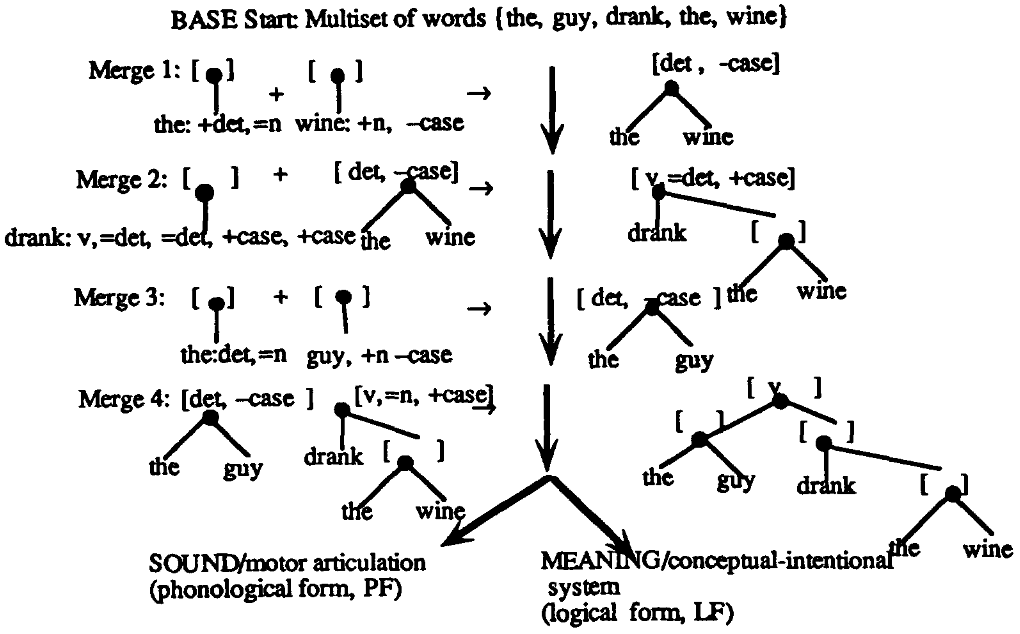

Fig. 3. Details of simple sentence derivation in the minimalist system. The basic derivational operator, Merge, applies four times, starting with an unordered world multiset. Each Merge combines either two words, indicated by straight lines, into a hierarchical superword, indicated by a triangle, or else combines two word/hierarchical superwords into a new hierarchical combination. Merger continues until all possible formal features have been eliminated.

1) Merge 3: combine *the* and *guy*, yielding *the guy*.

1) Merge 4: combine *drank the wine* and *the guy*, yielding *the guy drank the wine*.

Summarizing, Merge works on the model of chemical valency and feature cancelation. The core idea is that Merge takes place only in order to check features between its two inputs—a functor that requires some feature to be discharged, and an argument that 'receive' this discharged feature. The feature is then eliminated from further syntactic manipulation. After any Merge step, if a feature has not been 'canceled' by a functor–argument combination, that feature is 'copied' to the root of the combination and further Merges attempted until we are left with only phonological and logical form features. After exhausting all possible Merge sequences, if any non-phonological or LF features remain then the derivation is ill formed.

2.2.2. Minimalism and movement. So far we have described only how a simple sentence is derived. Following Kitihara, as described in Epstein [12], one can see that displacement or movement can be handled the same way, as a subcase of Merge. Figure 4 shows how. Suppose one forms the question, *What did the guy drink* by moving *what* from its canonical object position after the verb *drank*. Recall that we may define Merge as Merge(X,Y), where X and Y are either words or phrases. If X is a hierarchical subset of Y (roughly, a subtree), then this is a case of movement, as illustrated in the figure: X = *what* is a subtree of Y = the guy drink what. As usual, Merge forms a new hierarchical object, selecting and projecting one of the items, in this case *what*, as the root of the new 'tree'. As usual, we must assume that Merge is driven by feature checking: we assume that there is some feature, call it *Q* for 'question', that attracts

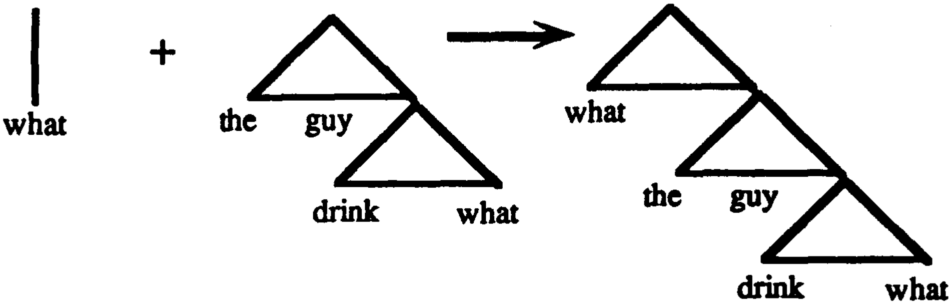

Fig. 4. Movement or phrase displacement as a subcase of Merge. In this figure, wh-question formation is depicted as the Merger of two elements, *what*, and a (traditionally named) sentence phrase, *the guy drink what*. Details about the inflection of *drink* and insertion of *do* to carry the inflection information are omitted.

what, while *what* has a -*Q* feature as before, *what* moves to the *closest* position where its feature may be checked. Note that movement now amounts to copying the displaced element to its new position, forming literally *what the guy drink what*. Presumably a general phonological principle at PF avoids 'pronouncing' *what* a second time, yielding the sentence that actually surfaces.

As we shall see in the next section, this approach also accounts for several of the formerly stipulated properties of movement. Perhaps more surprisingly, the notion of merge-as derivation suffices to fix precisely the syntactic relations appearing in natural languages, in this sense deriving a complex 'phenotype' from a much simpler 'genotype'.

2.3. Deriving syntactic relations and constraints from Merge

As described in the introduction, natural languages are characterized by certain specific properties, syntactic relations obtaining only among certain syntactic elements, under certain circumstances. These are evidently forced by the minimalist framework and Merge itself; let us review these here.

- *Recursive generative capacity*: this is a basic inherent property of Merge. Since Merge can apply recursively to its own output, indefinitely large hierarchical structures can be generated.
- *Structure dependence*: algebraically, Merge works via the concatenation of two (structured) objects. It is therefore a non-counting function: its inputs can be any two adjacent elements, but by definition it cannot locate the first auxillary verb *inside* a string of elements (unless that element happens to appear at the left or right edge of a phrase), nor, a *fortiori*, can it locate the third or 17th item in a string. Note that given a 'conceptually minimal' concatenative apparatus, this is what we should expect: clearly, Merge could not operate on a single argument, so the minimal meaningful input to Merge is two syntactic objects, not one or three.
- *Binary branching phrases*: since Merge always pastes together exactly two elements, it automatically constructs binary branching phrase structure.
- *Displacement* given Merge, the previous section showed that a mechanism to implement displacement exists. Again, whether and how a particular human language chooses to use displacement is an option dependent on the features of particular words (up to the constraints enforced by Merge). For example, English uses displacement to form wh-questions, given a *Q* 'attractor' in C(omplementizer) or root position, but Japanese does not.

If displacement is a subcase of Merge, then the following constraints on displacement follow—constraints that are all in fact attested.

- Displaced items *c-command* their original locations. C-command is the basic syntactic notion of 'scope' in natural language; for our purposes, c-command may be defined as follows [13]:

  A c-command B if and only if
  (1) The first branching node dominating A dominates B
  (2) A does not dominate B
  (3) A does not equal B

Figure 5 illustrates this. As one can see, in our displaced question sentence the first *what* (= A) c-commands the second *what* (= B), the object of the verb, because the first 'branching node' above *what* dominates (lies above) the second *what*. Note that the c-command relation is asymmetric: the second *what* does not c-command the first.

The c-command relation between displaced elements and their original locations *follows* from a general property of Merge: given any two inputs to Merge, X and Y, where X selects Y, then X c-commands Y and all the subcomponents of Y, because by the definition of Merge, we always form a new hierarchical structure with a root dominating *both* X and Y. In particular, for displacement, when X is a subcomponent (conventionally, a subtree) of Y, the displaced X must dominate the original location that is a subpart of Y. Below, we show how to derive the form that c-command takes from more primitive properties of Merge.

- *Lcoality conditions*: displacement is not totally free, because feature checking is local. What blocks question-formation such as *What do you know how the guy drank?*, while allowing *How do you know what the guy drank?* This too has a direct answer, given Merge. Note that any phrase such as *How the guy drank what* is 'locally convergent' in the sense that all its case and other feature-checking requirements have already been satisfied—this is what is called in lingusitics an adjunct phrase. In other words, *How* satisfies any feature-checking requirement for the full sentence's 'aspect'. Another way to think of the same situation is that at this particular point in the derivation only phonological and semantic features remain in this subphrase. Therefore, this phrase may already be shipped off to LF—spelled-out—and is thereby rendered opaque to further syntactic manipulation. If this is so, then there is nothing that allows *what* to participate in further Merges—that is, it can no longer be displaced or moved. In contrast, the hierarchical object corresponding to *did the guy drink*

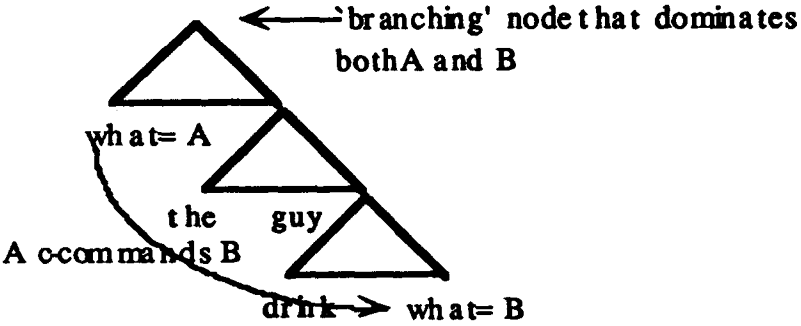

Fig. 5. C-command is an asymmetrical grammatical relation between two hierarchical nodes in a sentence structure. This example shows that displaced elements always c-command their original locations.

what is still open to syntactic manipulation, because (in English) the aspectual/question feature associated with the full sentence has yet to be satisfied—and in fact, can be satisfied by Merging *what* with the sentence, moving *what* to the front: *what did the guy drink what.* Finally, such a sentence may be combined as an arguement with *How do you know* to yield *How do you know what the guy drank.* In other words, given the local feature-checking driven properties of Merge, plus its operation on simply adjacent syntactic domains, we would expect locality roadblocks like the one illustrated.

To conclude our summary of how basic syntactic properties and relations can be derivable from the fundamental generative operator, following Epstien [12], we can demonstrate that natural languages can express only a limited set of relations like subject-of, object-of, and c-command.

For example, the c-command relation holds between the subject noun phrase *the guy* and the object *the wine*, but not vice versa. Why? In so-called representational theories of syntax, such as government and binding theory, the notion of c-command is given by definition [13]. Its exact formulation is stipulated. However, c-command is derivable from properties of Merge and the derivational formulation presented earlier, as are the other basic syntactic relations.

To see why, consider again the Merge operation. Merge takes a pair of syntactic object items and concatenates them. Syntactic structure is thus a temporal sequence of Merges, a *derivational history*. Given a derivational history and the sequence of sytactic structure the history traces out, we obtain the set of syntactically possible relations among syntactic objects. Let us see how. The derivation of our wine example is repeated below:

1) Merge 1: combine *the* and *wine*, yielding *the wine.*
2) Merge 2: combine *drank* and *the wine*, yielding *drank the wine.*
3) Merge 3: combine *the* and *guy*, yielding *the guy.*
4) Merge 4: combine *drank the wine* and *the guy*, yielding *the guy drank the wine.*

Now the notion of a possible syntactic object and relation can be expressed via the following definitions.

Definition 1
Let A be a *syntactic object* if and only if it is a selected word or a syntactic object formed by Merge.

Definition 2
A syntactic object is said to *enter in the derivation* if and only if it is paired with another object via Merge.

Definition 3
We say A and B are *connected* if they are parts of another (larger, common) syntactic object C.

We can now deduce c-command from Merge:

Theorem 1
Let A and B be syntactic objects. A *c-commands* B if A is connected to B at the step when A enters into the derivation.

Proof sketch. Without loss of generality, let us see how this works with our example sentence. When *the* and *wine* are merged, they both enter into the derivation, and thus either may c-command the other, as is required. Merge creates a new hierarchical object, essentially the projection of *the.* Analogously, the verb *drank* and the object (the traditional object noun phrase) *the wine* c-command each other, because *drank* is connected to *the wine* at the time of their merger. These are the straightforward cases. The property that is more difficult to see is how one can derive the asymmetry of c-command. For instance, *drank* also c-commands all the subparts of *the wine*, namely, *the* and *wine*, but *the* and *wine* do not c-command *drank*. This is because at the Merger step when *drank* entered the derivation it was *connected* to *the* and *wine*. But the converse is not true. At the time when *the* and *wine* entered into the derivation (when they were Merged to form *the wine*), *drank* was not yet part of the derivation, hense was not visible. Hence, *the* and *wine* do *not* c-command *drank*, as is required. Similarly, the subject phrase *the guy* c-commands *drank the wine* and vice versa—because these two objects are Merged. Letting A = *the guy* and B = *drank the wine*, we see that the subject noun phrase is by definition connected to all the subparts of *drank the wine* because it is connected to them at the time it enters the derivation. Therefore, th subject c-commands these subparts, as required. The converse is not true—neither *drank*, nor *the*, nor *wine* c-commands the subject—because for A = *wine* for instance, A was not connected to the subject *at the time* it entered into the derivation.

Indeed, it appears that if we take our definitions as specifying syntactic 'visibility', then all other syntactic relations reduce to subcases of the same criterion. Figure 6 illustrates the possibilites.

- *Object-of* is the relation: Merge and select a word base (functor) with either another word or a hierarchical structure.
- *Subject-of* is the relation: Merge a previously-merged hierarchical structure with a second hierarchical structure, selecting the first element as the new hierarchical root, and the second as the 'subject' (left-to-right order irrelevant—that is, the subject can appear either to the right or to the left).
- *Head-of* is the relation already described as the projection of features after Merge.
- No other (natural) syntactic relations are expected to be possible, e.g. *subject–object-of*, relating, say, *guy* to *wine*, since these items are not *connected* at the time of their mutual participation in Merge.

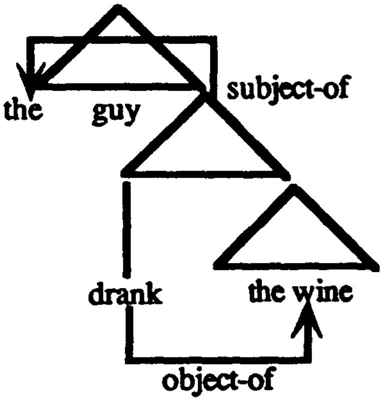

Fig. 6. The core syntactic relations are fixed by visibility at Merger time. This figure depicts the *subject-of* and *object-of* relations, with the 'selected' or functor-like pair of the Merger drawn as the source of the arrow.

2.4. Features and Merge: implications for dysphasia

With words and feature matching now cast as the central, perhaps the only component of syntax, it now seems possible to view all language deficits as the malfunctioning of some component of the word feature-Merger system. While it is not the point of this article to provide more than the barest description of these possiblities, it is worth sketching them below in table form, if only to make this point as well as to make the obvious suggestiong that a more fine-grained analysis of syntactic features might reveal more individual differences in performance related to these categories than has been previously suggested.

Failure mode/component	**Syntactic effect**	**Behavioral outcome**	**References**
Merger system	No recursion (phrase embedding)	Agrammatic aphasia	Geschwind (1979), Caplan (1987)[14, 15]
'Attract' feature	No 'traces'	Failure on agentive passives (''the boy is pushed by the man'')	Grodzinsky (1990) [16]
Feature matching/incorrect feature retrieval	Improper feature projection, matching	SLI	Gopnik and Crago (1990), Gopnik and Goad (1997), Rice (1994) [17, 5, 18]
Phonological/articulatory features	Defective PF	Perceptual/articulatory 'deficits'	Leonard (1994), Merzenich *et al.* (1996) [19, 20]
Semantic features	Incorrect thematic role assignment	'Fluent' aphasia	Caplan (1987) [15]

Of this array of dysphasias, most are familiar, with the possible exception of Grodzinsky's characterization of a certain class of agrammatic aphasias. Let us describe this briefly. Grodzinsky used a picture assembly task with a population of so-called agrammatical aphasics to account for two observations: first, that this population could still detect certain ungrammaticalities (as had been noted by others), for example, they knew that noun–noun combinations could not be 'merged' (using the terminology of this article); second, that they failed at tasks that required construction of a syntactic form that included traces, so that thematic roles could be correctly assigned (using the terminology of the older transformational grammer). Thus for example, he used passives in which either the subject or object could serve as the 'agent' of the sentence, as in the example in the table, so that the correct assignment of thematic roles would hinge on correct syntactic performance and not a general cognitive strategy, say, 'assign the first noun phrase as the agent of the sentence'. Grodzinsky found that indeed this population of individuals performed at chance level with such examples. In our terminology, such individuals could lack either the proper 'attraction' features (like the abstract Q morpheme in English questions) or 'Attract f' itself. The result in the former case would be a 'spotty' evidence of movement, while the loss of 'Attract f' would amount to the complete absence of movement, from passives to wh-questions and topicalization. Additional evidence for deficits of this kind come from intriguing data of Kegl (personal communication), who has discovered a ''double dissociation'' in aphasic American Sign Language (ASL) speakers with respect to wh-question formation. Specifically, in American Sign Language wh-questions are marked with certain facial and shoulder gestures—a quizzical rise of eyebrows, and a shoulder tilt. However, here there was a double dissociation: Kegl found both cases where ASL speakers had lost the ability to raise their eyebrows when asked to produce a wh-question, but had retained their ability to make the same gesture emotively, and vice versa, that is, cases where ASL speakers had lost their expressive ability but not their linguistic ability to raise their eyebrows. The connection to the feature-based Merge account is suggestive, but remains to be investigated in detail. One might expect that as the notion of 'feature' is further deepened that more subtle variations like this would be uncovered.

3. FROM MERGE TO LANGUAGE USE

A Merge-based model also meets a 'psychological fidelity' requirement for efficient language processing and accurate 'breakdown' processing, beyond the broader kinds of language breakdown just described. There is a natural, transparent relation between a Merger sequence and the operation of the most general kind of deterministic, left-to-right language analyzer known in computer science, namely, the class of LR parsers or their relatives, as we demonstrate below. In other words, given that the general hierarchical Merge operator forms the basis for natural language syntax, then an efficient processor for language follows as a by-product, again without the need to 'add' any new components. Of course, as is well known, this processor, like any processor for human language, has 'blind spots'—it will fall in certain circumstances, such as garden path sentences like *the boy got fat melted.* However, we can show that these failings are also a by-product of the processor's design, hence indirectly a consequence of the Merge machinery itself. In any case, these failings do not seem to pose an insuperable barrier for communicative facility, but rather delimit an envelope of intrinsically difficult-to-process expressions that then one tends to avoid in spoken or written speech [21].

First let us sketch the basic relationship between Merge and efficient LR parsing; see Berwick and Epstein [22] and Stabler [10, 11] for details and variations on this theme. The basic insight is simple, and illustrated in Fig. 7: a merge sequence like that in the figure mirrors in reverse the top–down expansion of each Rightmost hierarchical phase into its subparts. Thus, since parsing is the inverse of top–down generation, it should be expected to follow nearly the same Merger sequence 1–4 as in Fig. 7 itself, and it does. Consequently, all that is required in

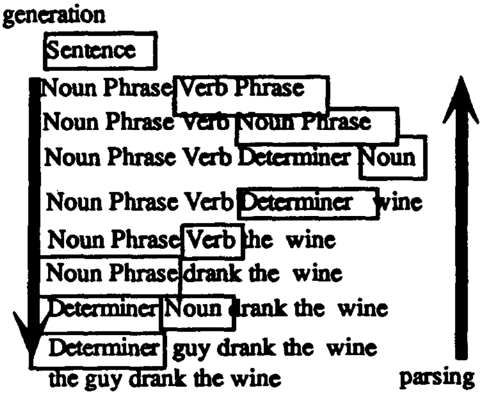
generation
Sentence
Noun Phrase Verb Phrase
Noun Phrase Verb Noun Phrase
Noun Phrase Verb Determiner Noun
Noun Phrase Verb Determiner wine
Noun Phrase Verb the wine
Noun Phrase drank the wine
Determiner Noun drank the wine
Determiner guy drank the wine
the guy drank the wine
parsing

Fig. 7. LR parsing is the mirror image of a top–down, right-most sentence derivation, and mirrors the Merge sequence for a sentence. This figure shows a line-by-line derivation for *the guy drank the wine*, where the boxed portion of each line shows that we expand the right-most possible portion at each step in a top–down generation. Naturally, in a bottom–up parse, we reverse this process, and recover the left-most complete hierarchical structure (the boxed portion) at each step.

order to parse strictly left to right, working basically bottom–up and building the Leftmost complete subtree at a time, is to reconstruct almost exactly the Merger sequence that generated the sentence in the first place. We assume in addition that if there is a *choice* of actions to take, then the processing system will again mirror the grammar, and so favor the 'economy' condition that the closest adjacent feature should be checked, rather than delaying to a later point in the derivation.

Such a parser can work with a simple push-down stack, and has just two possible operations: either *shift* a word (a feature bundle) onto the stack, analyzing its features; or *reduce* (that is, Merge) the top two items on the stack, yielding a new hierarchical structure that replaces these items on the stack, forming what is traditionally known as a 'complete subtree'. Figure 8 shows the blow-by-blow action of such a machine operating on our example *drank* sentence.

As each complete subtree is produced, we envision that it is shipped off to the conceptual/intentional component for interpretation, up to the as-yet-unmet features still left at the top of each hierarchical structure. Recall that this is possible for 'locally convergent' categories. For example, after *the* and *guy* are analyzed and the Merged, all the internal features of *the* and *guy*

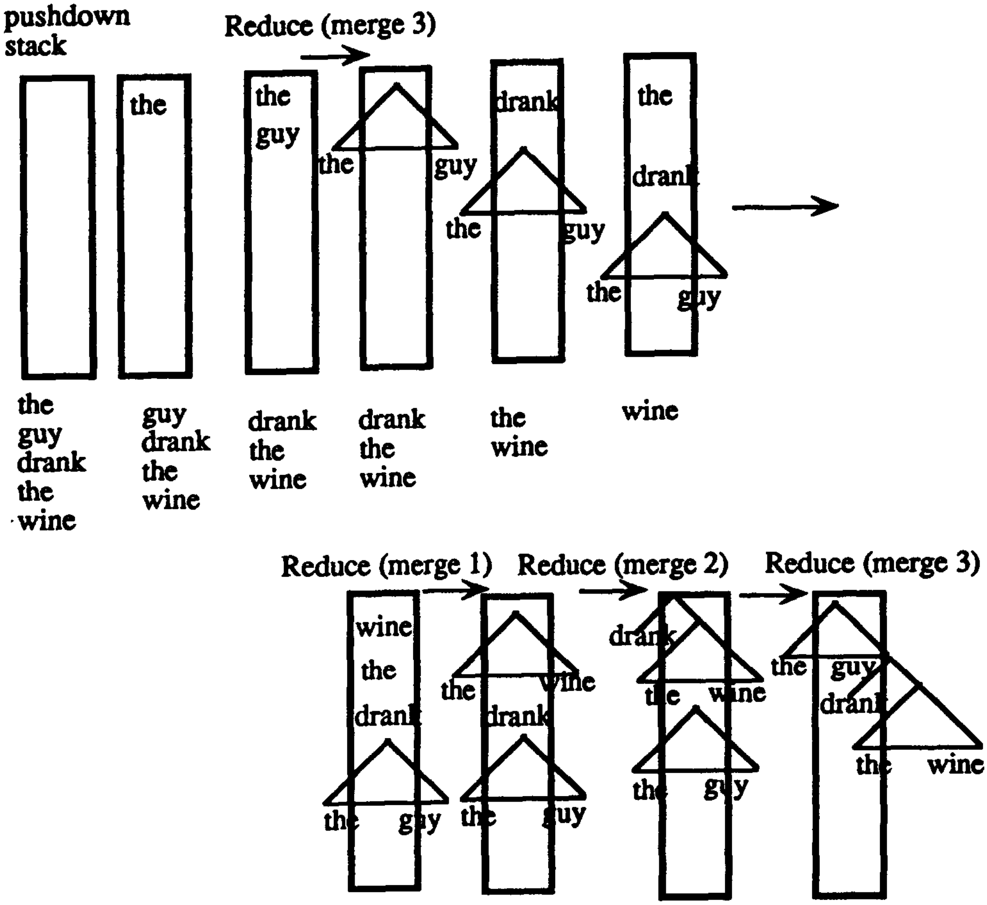

Fig. 8. This figure shows how a step-by-step LR parse for the sentence *the guy drank the wine* mirrors Merge steps 1–4 for the same sentence. Each 'reduction' corresponds to a Merge step.

have now been accounted for, aside from those that play a role *external* to the entire phrase, such as the phrase's thematic role—but these are precisely any features that have not yet been 'canceled' and are percolated to the head of the phrase for further processing. Thus these individual words may be interpreted. This proposal of incremental interpretation is essentially that found in Berwick and Weinberg [23].

We can trace through the parse of our example sentence in detail as follows, relating Merge to parsing actions.

Step 1. Shift *the* onto the stack, recovering its features from the lexicon. (From now on, we shall omit the phrase 'recovering its features from the lexicon' for each shift.)

Step 2. Shift *guy* onto the stack, on top of *the*.

Step 3. Merge 1: combine *the* and *guy*. Parser action: reduce *the* and *guy* to a complete phrase (left-most complete subtree) replacing *the* and *guy* on top of the stack with any uncanceled, projected features.

Step 4. Shift *drank* onto the stack.

Step 5. Shift *the* onto the stack.

Step 6. Shift *wine* onto the stack.

Step 7. Merge 2: combine *the* and *wine* into a new hierarchical object, replacing both on the stack (this is the object of the sentence). Parser action: reduce.

Step 8. Merge 3: combine *drank* and the object into a new hierarchical structure, traditionally known as a verb phrase, *drank the wine*. Parser action: reduce .

Step 9. Merge 4: combine *the guy* and *drank the wine* into a complete sentence. Parser action: reduce. The parse is now complete.

In many cases the choices for either shift or reduce (Merge) are deterministic, and allow such a device to work in the fastest possible time, namely, linearly in the length of the input sentence; but as is well known in order to handle the ambiguity present in natural language, we must generalize an LR machine to work 'in paralled' simply by carrying along multiple possibilities; there are known efficient algorithms for this [24]. In other cases, choices can be resolved by appeal to the 'local feature checking' or 'economy' condition imposed by the grammar; this leads directly to an account of some known language processing 'blindspots'. Consider as one example the reflexive attachment of *yesterday* in sentences such as *John said that the cat will die yesterday*, cited in the introduction. Why does the human sentence processor work this way? If in fact Merge proceeds by the 'most local' feature cancelation at each step, then the answer is clear: *yesterday* can be merged with the lower verb *die*, so this choice is made rather than waiting for so-called 'late attachment'—and this occurs before the *die* verb complex is shipped off for semantic interpretation. Hence, this is an operation that should be impervious to semantic effect, as indeed it seems to be. Similarly, such an approach also accounts for familiar cases of 'garden path' sentences, such as *the boy got fat melted.* Here too the basic situation, putting to one side many complexities, is that the noun–verb combination *boy got* is Merged ''too soon'' and taken as the main sentence—a processing error that we attribute to the 'local' character of feature matching. It remains to be seen whether all psycholinguistic 'blindspots' of this kind can be accommodated in the same way.

4. CONCLUSIONS

Taking stock, we see that Merge covers much ground that formerly had to be assumed in traditional transformational generative grammar. Many fundamental syntactic particulars are derivative: basic skeletal tree structure; movement rules; grammatical relations like object-of;

locality constraints; even the 'cyclic' character of grammatical rules—all these fall into place once the fundamental generative operation of Merge is up and running. These features are no less than the broad-brush outlines for most of human syntax—so nothing here has to be specifically 'selected for' in a gradualist, pan-selectionist sense. If so, then syntactic dysphasias should be expected to center around word matching and Merge as well, as seems to be true of the results emerging from much current research.

Of course, Merge will have little or nothing to say about the details of word features particular to each language—why English has a question word that sounds like *what*, or why such a word in English has features that force it to agree with an abstract question marker, while this is apparently not so in Japanese. Similarly, Chinese has no overt markings for verbal tense. The different words and associated features each language chooses ultimately lead to different possibilities for 'chemical combinations', hence different 'chemical compounds' or sentence construction types. But there is no need to invoke an array of distinct rules for each language, just as there is no need to invoke different laws of chemistry, once the basic principles are known. As Chomsky [1] has remarked, echoing the structuralists, while 'universal grammar' has a long history, nobody has ever assumed there would be a 'universal morphology'. Different languages will have different words with different features, and it is precisely here, where variation has been known all along, that languages would be expected to vary. In this sense, there is no possibility of an 'intermediate' language between a non-combinatorial syntax and full natural language syntax—one either has Merge in all its generative glory, or one has effectively no combinatorial syntax at all, but rather whatever one sees in the case of agrammatic aphasics: alternative cognitive strategies for assigning thematic roles to word strings. Naturally, in such a system that gives pride-of-place to word features, one would expect that deficits in feature recognition or processing—the 'feature blindness' described by Gopnik [5]—could lead to great cognitive difficulties; many important details remain to be explored here. But if the present account is on the right track, while there can be individual words, in a sense there is only a single grammatical operation: Merge. Once Merge arose, the stage for natural language was set. There was no turning back.

Acknowledgements—This research was supported by NSF grant 9217041-ASC and ARPA under the HPCC program. Noam Chomsky, Samuel Epstein, Charles Yang and Morris Halle provided many valuable comments; all remaining errors are our own.

REFERENCES

1. Chomsky, N. A., *The Minimalist Program*. MIT Press, Cambridge, MA, 1995.
2. Chomsky, N., Remarks on features and minimalism. In *Is the Best Good Enough?*, ed. D. Pesetsky. MIT Press, Cambridge, MA, 1966.
3. Gopnik, M., Feature-blind grammar and dysphasia. *Nature* **344,** 715, 1990.
4. Gopnik, M., Dalalakis, J., Fukuda, S. E., Fukuda S., and Kehayia, E., Genetic language impairment: unruly grammars. In *Evolution of Social Behaviour in Primates and Man*, eds W. G. Runciman, John Maynard Smith and R.I.M. Dunbar, pp. 223–249. Oxford University Press, Oxford, 1996.
5. Gopnik, M. and Goad, H., What underlies inflectional error patterns in genetic dysphasia? *Journal of Neurolinguistics*, **10,** 109–137, 1997.
6. Beadle, G. W., *The Language of Life; An Introduction to the Science of Genetics*. Doubleday, Garden City, NY, 1966.
7. Jenkins, L., *Biolinguistics*. Cambridge University Press, New York, 1997.
8. Fodor, J., Bever, T. and Garrett, M., *The Psychology of Language*. McGraw-Hill, New York, 1974.
9. Chomsky, N. A., *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965.
10. Stabler, E., *Minimalism and Sentence Processing*. CUNY Sentence Processing Conference, New York, 1996.
11. Stabler, E., Parsing and generation for grammars with movement. In *Principle-based Parsing: From Theory to Practice*, ed. R. Berwick. Kluwer, Dordrecht, 1996.

12. Epstein, S.D., *Un-principled Syntax and Derivational Relations*. Harvard University, 1995.
13. Reinhart, T., *Anaphora and Semantic Interpretation*. University of Chicago Press, Chicago, 1978.
14. Geschwind, N., Specializations of the human brain. *Scientific American*, Sept. 1979.
15. Caplan, D., *Neurolinguistics and Linguistic Aphasiology*. Cambridge University Press, New York, 1987.
16. Grodzinsky, Y., *Theoretical Perspectives on Language Deficits*. MIT Press, Cambridge, MA, 1990.
17. Gopnik, M. and Crago, M., Familial aggregation of a developmental language disorder. *Cognition* **39,** 1–50, 1991.
18. Rice, M. L., Grammatical categories of children with specific language impairments. In *Specific Language Impairments in Children*, eds R. V. Watkins and M. L. Rice. Paul H. Brookes, Baltimore, 1994.
19. Leonard, L., Some problems facing accounts of morphological deficits in children with specific language impairment. In *Specific Language Impairments in Children*, eds R. V. Watkins and M. L. Rice. Paul H. Brookes, Baltimore, 1994.
20. Merzenich, M., Jenkins, W., Johnston, P., Schreiner, C., Miller, S. and Tallal, P., Temporal processing deficits of language-learning impaired children ameliorated by training. *Science*, **271,** 77–83, 1996.
21. Chomsky, N. A. and Miller, G. A., Finitary models of language users. In *Handbook of Mathematical Psychology*, eds R. Luce, R. Bush and E. Galanter, pp. 419–491, 1963.
22. Berwick, R. C. and Epstein, S. D., Merge: the categorial imperative. In *Proceedings of the 5th AMAST Conference*, University of Twente, Twente, December 1995.
23. Berwick, R. C. and Weinburg, A. S., *The Grammatical Basics of Linguistic Performance*. MIT Press, Cambridge, MA, 1984
24. Tomita, M., *Generalized LR Parsing*. Kluwer, Dordrecht, 1986.