Exploring Crash-Proof Grammars

Edited by Michael T. Putnam

Language Faculty and Beyond
Internal and External Variation in Linguistics
3

John Benjamins Publishing Company

Exploring Crash-Proof Grammars

Edited by

Michael T. Putnam

The Pennsylvania State University

John Benjamins Publishing Company

Amsterdam / Philadelphia

The paper used in this publication meets the minimum requirements of American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

Library of Congress Cataloging-in-Publication Data

Exploring crash-proof grammars / edited by Michael T. Putnam.

p. cm. (Language Faculty and Beyond, ISSN 1877-6531 ; v. 3)

Includes bibliographical references and index.

1. Grammar, Comparative and general--Syntax. 2. Generative grammar. 3. Minimalist theory (Linguistics) I. Putnam, Michael T.

P291.E98 2010

415--dc22 2010018680

ISBN 978 90 272 0820 0 (Hb ; alk. paper)

ISBN 978 90 272 8801 1 (Eb)

John Benjamins Publishing Co. · P.O. Box 36224 · 1020 ME Amsterdam · The Netherlands

John Benjamins North America · P.O. Box 27519 · Philadelphia PA 19118-0519 · USA

Preface & acknowledgments

The idea for this volume, as well as the source for the majority of these papers, stems from the *Exploring Crash-Proof Grammars Conference* held at Carson-Newman College on February 29-March 2, 2008. This conference was the ideal setting to exchange ideas and views on theories and perspectives related to the concept of crash-proof grammars. The participants of this conference created an environment conducive for the exchange of new ideas. It was a privilege to be a part of such an academically stimulating conference. Of course, none of this would have been possible without the financial support of the School of Humanities at Carson-Newman College as well as the time and effort of my colleagues and friends in both the Departments of English and Foreign Languages. I would also like to recognize Kleanthes Grohmann and Pierre Pica for the invitation to submit this volume for review in this excellent series. Again, I am deeply indebted to all of you for your vision and support. Thanks again.

University Park, PA
USA
August 2010

List of contributors

Cedric Boeckx
Centre de Lingüística Teòrica
Universitat Autònoma de Barcelona
Departament de Filologia Catalana
Edifici B
Campus UAB
08193 Bellaterra (Barcelona)
Spain
cedric.boeckx@uab.cat

Hans Broekhuis
Meertens Institute
P.O.Box 94264
1090 GG Amsterdam
The Netherlands
hans.broekhuis@meertens.knaw.nl

Vicki Carstens
University of Missouri-Columbia
Department of English
316 Tate Hall
Columbia, MO 65211
USA
carstensv@missouri.edu

Rui Chaves
SUNY @ Buffalo
Department of Linguistics
627 Baldy Hall
Buffalo, NY 14260-1030
USA
rchaves@buffalo.edu

Samuel D. Epstein
University of Michigan
Department of Linguistics
411 Lorch Hall
611 Tappan Street
Ann Arbor, MI 48109-1220
sepstein@umich.edu

Dennis Ott
Harvard University
Department of Linguistics
Boylston Hall, 3rd Floor
Cambridge, MA 02138
USA
dott@fas.harvard.edu

Hamid Ouali
University of Wisconsin-Milwaukee
Department of Linguistics
P.O. Box 413
Milwaukee, WI
USA
ouali@uwm.edu

Michael T. Putnam
The Pennsylvania State University
Department of Germanic & Slavic
Languages & Literatures
427 Burrowes Building
University Park, PA 16802
USA
mtp12@psu.edu

T. Daniel Seely
Eastern Michigan University
Department of English/Linguistics Program
612D Pray-Harrold Hall
Ypsilanti, MI 48197
USA
tseely@emu.edu

Halldór Sigurðsson
Lund University
Centre for Language & Literatures
Box 201
221 00 Lund
Sweden
halldor.sigurdsson@nordlund.lu.se

Hisatsugu Kitahara
Keio University
Institute of Cultural & Linguistic Studies
2-15-45 Mita, Minato-ku
Tokyo 108-8345
Japan
kitahara@icl.keio.ac.jp

Tommi Tsz-Cheung Leung
United Arab Emirates University
Department of Linguistics
P.O. Box 17771
Al-Ain
U.A.E.
leung@uaeu.ac.ae

Joan Maling
Brandeis University
Department of Psychology
MS 062 P.O.Box 549110
Waltham, MA 02454
USA
maling@brandeis.edu

Thomas Stroik
University of Missouri-Kansas City
Department of English
Kansas City, MO 64110
USA
stroikt@umkc.edu

Balázs Surányi
Research Institute for Linguistics of the
Hungarian Academy of Sciences
Zichy u. 20. III./3.
1062 Budapest
Hungary
suranyi@nytud.hu

Ralf Vogel
Universität Bielefeld
Fakultät für Linguistik &
Literaturwissenschaft
Postfach 10 01 31
33501 Bielefeld
Germany
ralf.vogel@uni-bielefeld.de

Table of contents

Exploring crash-proof grammars

An introduction*

Michael T. Putnam
The Pennsylvania State University

1. Introduction

The central aim of any linguistic theory is to develop a restrictive, explanatory theory of the properties of all human languages. In the Minimalism Program (MP), Chomsky (1993, 1995, and subsequent work) moves away from devising rule-system analyses of grammar and returns to his original objective of providing a description of the cognitive state of the human mind and its relation to Human Language. Under minimalist assumptions, Human Language consists of a Computational System that derives structured representations (i.e. grammatical strings) to external performance systems (commonly referred to as the Conceptual-Intentional (C–I) and Sensori-Motor (S–M) interfaces) for full interpretation (FI). Minimalism interprets human language as being perfect in design and as an optimal format for producing structures that can be used and interpreted by these external interfaces (cf. (C–I) and (S–M)).[1] On the grounds of such a strong minimalist hypothesis, the notion that a theory of the syntax of natural language

*I would like to extend a special thanks to the participants of the *Exploring Crash-Proof Grammars* conference held at Carson-Newman College (February 29–March 2, 2008). Many of the ideas and thoughts that were shared and discussed among those in attendance as well as the participants helped shape not only this introduction, but also the volume as a whole. I am also deeply indebted to the following individuals for more in-depth discussions and conversations concerning this introduction: Hans Broekhuis, Rui Chaves, Sam Epstein, John Hale, Kyle Grove, Sam Epstein, Daniel Seely, Tom Stroik, Takashi Toyoshima, John te Velde and two anonymous reviewers. Any inconsistencies and shortcomings in this introduction are not the fault of these individuals and thus rest solely on the shoulders of the author.

1. The idea that language – as a biological system – is "perfect" in design is not uncontroversial. Johnson and Lappin (1999) suggest that biological systems are often not "optimal" in design, let alone "perfect".

could be void of fatal errors – at least as far as external interface interpretation is concerned – is worthy of serious exploration.

As with the rule systems of early generative grammar, the Computational System of Human Language (C_{HL}) in the Minimalist Program is designed to protect itself from illicit (uneconomical) derivations and interface-unusable representations. Deviant derivations and representations, according to Chomsky, will **crash**:

> Invariant principles determine what counts as a possible derivation and a possible derived object (linguistic expression, SD). Given a language, these principles determine a specific set of derivations and generated SDs, each a pair (π, λ). Let us say that a derivation D converges if it yields a legitimate SD and crashes if it does not; D converges at PF if π is legitimate and crashes at PF if it is not; D converges at LF if λ is legitimate and crashes at LF if it is not. In an EST framework, with SD = (δ, σ, π, λ) (δ a D-Structure representation, σ an S-Structure representation), there are other possibilities: δ or σ, or relations among (δ, σ, π, λ), might be defective. Within the Minimalist Program, all possibilities are excluded apart from the status of π and λ. A still sharper version would exclude the possibility that π and λ are each legitimate but cannot be paired for UG reasons. Let us adopt this narrower condition as well. Thus, we assume that a derivation converges if it converges at PF and at LF; convergence is determined by independent inspection of the interface levels – not an empirically innocuous assumption (…) (1995: 171)

> The Language L determines a set of derivations (computations). A derivation converges at one of the interface levels if it yields a representation satisfying FI at this level, and converges if it converges at both interface levels, PF and LF; otherwise, it crashes. We thus adopt the (non-obvious) hypothesis that there are no PF-LF interactions relevant to convergence – which is not to deny, of course, that a full theory of performance involves operations that apply to the (π, λ) pair. Similarly, we assume that there are no conditions relating lexical properties and interface levels, such as the Projection Principle. The question of what counts as an interpretable legitimate object raises non-trival questions, some discussed in earlier chapters (…) (1995: 220)

This interpretation of the definition of **crash** has remained largely intact, at least in the works of Chomsky since the onset of the MP, as illustrated by the following quote:

> If these general ideas are on the right track, then all internal levels are unformulable, hence dispensable in the strongest sense. We are left with only the interface levels, and the five cycles of the EST/Y-model are reduced to a single one, based on Merge. The cyclic properties of the mappings to the interface follow without comment. (2006: 11)

As formal theories of syntax have moved further away from purely structural descriptions and more towards an account of the biological/cognitive state of the human mind, it becomes less clear if the previous working definitions of **crash** can be maintained and, more importantly, exactly what they signify in either an intensional or extensional sense. As we now revisit why principled and parameterized structures in human language exist as they do, we could potentially make some deeper sense of the structures of language and conjecture that they could potentially reflect – at least to some degree – a *perfect design*. Although originally discussed by Chomsky (1995), Frampton & Gutmann's (2002) seminal work on a syntactic theory perfect in design and void of **crashes**, hence coining the term **crash-proof** in the process, completely removes the notion of **crash** from the formal syntax and situates it at external units (i.e. LF and PF). This discussion, of course, lies at the very heart of this volume of papers dedicated specifically to exploring the idea of the characteristics that a crash-proof grammar would exhibit and if such a system is truly possible (and, in the end, desirable) in the first place.

2. Defining crash(es)

Epstein, Kitahara & Seely (this volume: 1) make the following observations about the concept **crash** and its complement **converge**: "a derivation D converges at an interface level if and only if (iff) the interface can actually interpret each and every feature appearing in the expression generated by D. Expressions must be 'useable,' and they are useable (by hypothesis) iff they contain only legible features in certain arrangements. Derivations failing to generate such expressions crash." These observations are consonant with Chomsky's (2000: 95) claim that "A computation of an expression Exp converges at an interface level IL if Exp is legible at IL, consisting solely of elements that provide instructions to the external systems at IL and arranged so that these systems can make use of them; otherwise, it crashes at IL. The computation converges if it converges at all interfaces."

Given the above remarks as our starting point, we can define a **crash**, in its most basic form, as in (1):

(1) **Crash** (to be revised):

If a syntactic object α cannot be interpreted at an IL in any and all of its features, α is neither useable nor legible at IL.

Although the working definition for **crash** seems straightforward enough, many unanswered questions remain. To illustrate this point, consider the following strings:

(2) a. He fired her.
    b. *He cried her.

The example (2b) is a clear violation of the Theta Criterion/Full Interpretation since the predicate **cry** fails to license and assign a theta-role to its internal object

her. The difficulty arises in determining when and where the Theta Criterion/Full Interpretation – which is assumed to be present and active at the C–I interface – is violated in the concatenation of the syntactic object *[$_{VP}$ cried her]. Does the evaluation *[$_{VP}$ cried her] take place immediately after the concatenation of [$_{V}$ cried] and [$_{DP}$ her] or at a later stage in the derivation? In a related sense, consider the sentence in (3):

(3) Rats like cheese.

As discussed by Epstein (2007) and Epstein & Seely (2006), the derivation of (3) should involve a "**crash**" because the derivation will create the syntactic object [$_{VP}$ like cheese] and this VP will violate the Theta Criterion (since the predicate *like* has two theta-roles to assign and in this syntactic unit the theta-role for the external argument (*rats*) has not yet been assigned). However, this "**crash**" is not fatal, i.e. it does not force a termination of further derivational operations built upon this "crashed" VP.[2] To differentiate the Theta Criterion violation (and subsequent **crash**) in (2b) from the one in (3), Epstein & Seely (2006) and Epstein (2007) propose an elaborated sense of the concept **crash**, one which does not buy into the strong hypothesis that all crashes produce a derivational unit that cannot be used/interpreted by the interfaces. Epstein & Seely claim that one of the key issues preventing us from properly understanding and classifying these sorts of non-fatal crashes is our continued reliance upon the Government and Binding (GB) characterization of the notion of 'grammatical sentence' (see Chomsky (1981)). Consider the following GB assumptions about well-formedness (taken from Epstein & Seely 2006: 179):

(4) a. All and only grammatical sentences have 'well-formed' derivations.
    b. A derivation is well-formed only if at every point in the derivation no principle is violated.

Epstein & Seely (2006: 179) argue that if the external interfaces LF and PF must access/interpret every successive derivational step and if the criteria for well-formedness are as stated in (4), then "no well-formed derivations of grammatical sentences would ever be generable." We can see this in (3), where deriving the initial VP [like cheese] violates the Theta Criterion/Full Interpretation and, given the conditions in (4), the derivation for (3) should immediately **crash**. What Epstein & Seely conclude from this state of affairs is simply that some violations must be

2. This is only problematic if one assumes a level-free derivational system that assumes that LF necessarily accesses each syntactic object at each point in a derivation; LF does not wait until the 'end of the line' to interpret these syntactic objects.

tolerated in minimalist/derivational syntactic theory. Consequently, Epstein & Seely adopt the following amendments to the Computational System: each generated representation (i.e. syntactic object) has PF and LF properties; if α is non-convergent, it does not follow that β containing α (β a new object) will also **crash**. To illustrate this point, consider (5; see also ex. (18); Epstein & Seely (2006: 183)):

(5) 1st merge: like + cheese; PF and LF interpret → A unit derivation
2nd merge: rats + [like cheese]; PF and LF interpret → A unit derivation
A unit derivation + A unit derivation → An extended derivation

As illustrated in (5), even though the first merge of [$_{VP}$ [$_{V}$ like] [$_{DP}$ cheese]] yields a unit derivation that violates the Theta Criterion, the second application of Merge produces a syntactic object which satisfies this requirement. When interpreted together as an extended derivation, no violations of the Theta Criterion are observed. As a result, a derivation can proceed from a unit derivation that exhibits a **crash** to another unit that is well-formed, and, in essence, repairs the extended derivation.[3] If we follow this line of reasoning, we are forced to revise our original definition of **crash** in (1):

(6) **Crash** (version 2)

If a syntactic object α cannot be interpreted at an IL in any and all of its features, α is neither useable nor legible at IL, **iff α cannot be combined with another local[4] derivational unit that repairs the violation(s) of α.[5]**

3. According to Epstein & Seely (2006: 183), it is apparently possible to proceed derivationally from a syntactic unit that doesn't **crash** to a syntactic unit that does **crash**. This, however, would (as far as I can tell) always result in an ungrammatical extended derivation, which indicates that these are errors that human grammars would never produce. Such phenomena that come to mind in light of this point where such good-to-bad sub-derivations exist in a feeding relationship are instances of improper movement and islands. However, it may be the case in a system that is tolerant of non-terminal **crashes** that the only local derivational unit that must be pure of **crashes** is the final step where both local and global PF and LF can determine if the entire string is interpretable at the interfaces.

4. I apply the term "local" here very loosely, i.e. in a strict sense, one could interpret it to mean purely sisterhood (i.e. Spec-Head) relations. Contrariwise, another possible interpretation of "local" here could be the c-command relationship between α and β.

5. Another implication of this version of **crash** that isn't clear is whether or not the repairing derivation must immediately follow the non-fatal violation. If the repair must be immediate,

The revised definition of **crash** in (6), however, still faces many conceptual issues. First, it forces the theory to adopt both a local *and* a global interface interpreting mechanism, at least for LF in this instance and quite possibly for PF as well. Second, as I mention below (cf. fn. (5)), it is not clear whether the non-fatal crash realized at step Σ_n must be repaired immediately at step Σ_{n+1} (we'll return to this issue below). One would assume that this has to be the (optimal) case, or else the derivational system would be permitted to stack uninterpretable derivational units on top of one another, in the anticipation of a syntactic object/unit derivation that would repair all previous **crashes** in its extended derivation. Third, and in a related sense, Epstein & Seely only extend the novel sketch of non-fatal **crashes** to Theta Criterion violations (although see Epstein (2007) for extended discussion of other related issues). In order to determine whether or not we wish to support the existence of non-fatal crashes in a minimalist/derivational syntactic theory, other features and criterion must be examined.

What emerges from this discussion on the (potential) existence of non-fatal crashes in a derivational system is this: these multiple flavors of **crash** are necessary only if one assumes a Computational System that allows multiple Transfer/Spell-Out. If one assumes, following Chomsky (2000 and later work), that all the complements of every phase head Transfers to the interfaces, or if one assumes, following Epstein & Seely, that the derivational output of every syntactic operation Transfers to the interfaces, then we must have a Computational System that treats the **crashes** of non-terminal structures differently than the **crashes** of terminal structures. Since non-terminal structures will not be usable by the interfaces,[6] assuming multiple transfers to the interfaces will require one to also assume that there are multiple flavors of **crashes** in which some instances of **crash** can be repaired and thus the derivation saved. Needless to say, the fact that every non-terminal derivational unit $X_1 \ldots X_n$ shipped off to the interfaces will (in most

then three-place verbs could pose a problem because merging the first argument with the verb will yield a double violation of the Theta Criterion and this double violation cannot be repaired in the next derivational step. On the other hand, if the repair needn't occur in the next derivational step, how long into the derivation can the non-repaired **crash** go before it becomes a "fatal crash"?

6. Tom Stroik (p.c.) points out to me that if non-terminal structures were usable at the interfaces, then we should expect to find what he calls 'bottom abortings', such as (a).

a. … eats pizza

These structures must in some sense **crash**; however, as pointed out by Epstein & Seely, these **crashes** cannot be fatal, or else every derivation in minimalist theory will fatally crash and be uninterpretable at the interfaces.

instances) have some unchecked/unvalued feature means that derivations will be replete with crashes. This approach, of course, still has lingering aspects that require deeper query. One of the potential problems with such a crash-tolerant model of syntax is that no good explanation exists as to why it is necessary to send all these unusable/uninterpretable structures to the interfaces. To put it bluntly, what are the interfaces supposed to do with these structures? These structures are – as far as I can tell – unusable and useless to the interfaces; the interfaces themselves cannot repair them because such repair can only occur in the computational system; and the interfaces cannot send these structures back to the derivation, mainly because these iterative applications of Transfer and Return operations would be circular/redundant in nature and accomplish nothing. In fact, it seems that the only reason the Narrow Syntax jettisons these structures is to ensure that they are not syntactically reusable (i.e. to limit the look-back domain of Internal Merge). Be that as it may, we still have no clear reason behind the conceptual necessity of transferring these unusable structures to the interfaces. In a program that embraces the rubric of virtual conceptual necessity as a way of determining what mechanisms should remain or be eliminated from its theory, the operation Transfer should be subjected to the same scrutiny in the minimalist literature. One potential improvement to this system is to adopt Boeckx's (2007) elimination of Transfer and re-interpretation of Spell-Out as an operation where the interfaces can peer into the Narrow Syntax and interpret what is necessary without transferring derivational units to the interfaces. Although this is a conceptual improvement, the issue of figuring out what to do with crashes in the Narrow Syntax remains.

To recapitulate, recent studies suggest that there are multiple flavors of crashes; some being fatal whilst other being non-fatal. Stroik (2009a, 2009b and previous work) maintains that this view of the MP is one of a theory that must be *crash-tolerant* in its weakest state, and in some cases, one could argue that such a theory is actually *crash-driven.* Under such assumptions, the operation Merge must be reformulated. No longer is Merge, as has been traditionally postulated in Minimalism, an operation driven by feature-interpretation, rather Merge is "free." Such a notion is in harmony with a version of the MP that advocates a *crash-tolerant* derivational system. This idea, namely, that certain crashes are tolerable and necessary, has been the focus of much research about performance. In fact, the idea that competence grammar should be process-neutral and compatible with production/comprehension theories of performance (cf. Chomsky 1965), is one of the reasons why non-derivational frameworks such as Head-Phrase Structure Grammar (HPSG) and Lexical Functional Grammar (LFG) have evolved to be "crash proof", very much in line with traditional notion of "crash proof" syntax as outlined by Frampton & Gutmann and others.

In the end, any interpretation of what crash-proof means and how it could and should be implemented in the MP, of course, returns us to our lack of a lucid definition of **crash** in present-day Minimalism. In this introduction, I have briefly discussed the two most salient alternatives to a definition of **crash.** To summarize, we are presented with these two possible interpretations:

(7) a. **Strict crash**[7] (cf. (1)): If a syntactic object α cannot be interpreted at an IL in any and all of its features, α is neither useable nor legible at IL.

b. **Soft crash** (cf. (6)): If a syntactic object α cannot be interpreted at an IL in any and all of its features, α is neither useable nor legible at IL, **iff α cannot be combined with another local derivational unit that repairs the violation(s) of α.**

Regardless of whichever stripe of **crash** we pursue in a minimalist system, it stands to have a drastic impact on almost every aspect of the fundamentals of such a theory. By adopting the first option (cf. (7a)), namely **strict crash**, we inherit a Merge operation that is "meaningful" (most likely feature-driven), a strict version of locality (i.e. Spec-Head) with regard to measuring syntactic relations, and, most likely, the elimination of larger ontological commitments such as phases, PIC, Probe-Goal relations. In contrast, applying the latter notion of **crash** (cf. (9b)), namely **soft crash**, to the MP results in an interpretation of Merge that is "free" (i.e. one that can tolerate non-fatal crashes), a view of locality that includes the notions of c-command (both from a representational and derivational perspective), and the presence of phases, the PIC, and Probe-Goal relations in the C_{HL}.[8] Admittedly, the alternatives to **crash** that I have presented here are polar opposites; the possibility does exist that variations of these definitions along some sort of scale could be integrated in the MP.

As insinuated throughout this introductory discussion, our view of **crash** affects our stance on the properties of Merge, the definition and function of locality, the role of interface constraints (e.g. phases and the PIC) on partial and complete structures and their interface interpretability, and the necessity of the

7. In a technical sense, the notion of **strict crash** could easily be incorporated into **soft crash** due to its unspecified time frame with regard to ***when*** an IL and its features must be interface-interpretable.

8. Recent work by Bošković (2007) and Fox & Pestesky (2005) expresses that the notion of phases and the PIC do not constraint for syntactic operations and do not determine locality in a syntactic sense. Rather, phases and the PIC are best understood as ***phonological*** constraint present at PF.

Transfer/Spell-Out operation.[9] Lastly, we cannot rule out the possibility that another approach to a crash-proof version of natural language syntax may define **crash** as something else entirely different from what we have mentioned up until now; namely, as real and measurable processing errors in language use. Following Chomsky (1965) and others such as Sag & Wasow (2007) and Chaves (this volume), this view of processing-**crashes** are phenomena that ought to be included in a united view of language and subsequently ruled out in a model of the grammar.[10] Relevant to this discussion at hand, the version of **crash** that we eventually settle on will affect our interpretation of the notion of a crash-proof syntactic system and whether, in the end, the pursuit of such a conceptual idea is a necessary or desirable attribute of the design of C_{HL}. In essence, any discussion about **crashes** and the pros and cons surrounding crash-proof syntax and its necessity (or lack thereof) permeats all aspects of the conceptual design of formal linguistic theories of Human Language.

3. Scope and content of this volume

The contributions to this volume fall into two camps. The first group of papers, found in *Part I: Applications of Crash-Proof Grammar*, provides analyses couched within some version of a crash-proof grammar, or at the very least, in a formalism that addresses the concept of crash-proof. Hamid Ouali and Vicki Carstens engage in discussions centering on the role of Feature Inheritance and Probe-Goal relations at both the CP- and DP-level. Looking at anti-agreement effects in Berber and gender features in Romance and Bantu languages, Ouali and Carstens respectively and independently provide analyses of how the notion of Feature Inheritance finds a natural home in any discussion involving crash-proof grammars.[11]

9. Another important issue related to the notion of **crash** employed in a variant of the MP that I will not address in any detail in this introduction is the level of importance afforded to the structure of *v*P (in particular, but application to phrase structure in general) (see Borer (2003, 2005) for an exhaustive discussion of this exo-skeletal approach to phrase structure) versus the features present on lexical items in the Lexicon. The treatment of these issues is taken up by Stroik (2009a, 2009b).

10. The infamous problem with such an approach is the blurring of the knowledge-performance distinction. For a more detailed discussion of this and related issues, see Chaves (this volume).

11. As pointed out to me by Hamid Ouali (p.c.), not all versions of crash-proof grammar abandon Probe-Goal relations (i.e. *Agree*). As a matter of fact, *Agree* plays a dominant role in Frampton & Gutmann's (2002) seminal discussion of crash-proof syntax (i.e. their notion of 'pivot'). There are, of course, some instantiations of strongly derivational theories that maintain that *Agree* should not be a component of a crash-proof syntax (cf. Putnam & Stroik (this volume)).