Radical: A Journal of Phonology
ISSN 2592-656X

volume 1, 2020

Scheer, Tobias

# On the lexical character of intermodular communication

Scheer, Tobias. 2020. "On the lexical character of intermodular communication". *Radical: A Journal of Phonology*, 1, 183-239.

Editor: *Guillaume Enguehard*
Reviewers: *Gabriel Bergounioux, Florian Breit, Eugeniusz Cyran*



PAPER

# ON THE LEXICAL CHARACTER OF INTERMODULAR COMMUNICATION

SCHEER, TOBIAS (UNIVERSITÉ CÔTE D'AZUR)

All theories of the morpho-syntax – phonology interface at least since the emergence of the generative approach practice two-channel translation, i.e. the conversion of morpho-syntactic structure into phonological units by two distinct mechanisms, one lexical (or list-type: morphemic information stored in the lexicon, i.e. Vocabulary Insertion), the other computational (boundary information, i.e. #, units of the Prosodic Hierarchy etc., which are not stored). It is argued that computational translation does not qualify for intermodular communication because computational systems that can read two distinct vocabularies (of the sending and the receiving module, what Jackendoff calls bi-domain specificity) are modular monsters. They defy the purpose of modularity (computational systems are not all-purpose but domain-specific, i.e. can only process information belonging to their own domain) and make interface devices pointless: if modules can parse the vocabulary of their neighbors, no translation is needed in the first place. It is therefore suggested that the translation of both morphemic and boundary information must be lexical. Within language (phonology – phonetics interface) and beyond, the generalization of lexical translation as the regular interface of a module-based cognitive system is then examined in a second step: all modules possess a lexicon on their input side which translates variable inputs into the uniform to-be-computed vocabulary. Objections against lexical translation that are raised on the Faculty of Language blog are discussed (e.g. is lexical translation the odd man out in the cognitive system?). Finally, it is argued that the translation of real-world items into cognitive categories (e.g. the association of wave lengths with colors perceived) is also list-based. Lexical translation thus appears to qualify as the general mechanism that manages intermodular communication (internal to the cognitive system) as well as the relationship between cognitive categories and the real world.

Modularity, morpho-syntax – phonology interface, phonology – phonetics interface, transduction, multisensory integration.

## PURPOSE

At least since SPE, all theories of the morpho-syntax – phonology interface practice two-channel translation,[1] i.e. the conversion of morpho-syntactic structure into phonological units by two distinct mechanisms, one list-type (morphemic information stored in the lexicon), the other computational (boundary information, i.e. #, units of the Prosodic Hierarchy etc., which are not stored). It is argued below that intermodular translation can only be non-computational, i.e. achieved by a lexical access as we know it from morphemic information (Vocabulary Insertion). Computational translation does not qualify because it is a modular monster: it requires the processing of information from two distinct domains (or vocabularies) pertaining to the two modules that are related. This defies the purpose of modularity, though, which is the insight that computational systems are not all purpose but domain-specific, i.e. process only information

1 In this article no distinction is made between translation and transduction. The latter is sometimes used in order to specifically refer to communication between the cognitive system and the real world beyond the skull.

belonging to their own domain. What Ray Jackendoff calls bi-domain specificity, i.e. a computational system processing information from two distinct domains (or vocabularies), is thus impossible on modular standards.

Two consequences of lexical translation are discussed in section 6. The output of translation must be storable (a good lexical entry), and the relationship between the two items associated by a lexical entry is arbitrary (this is the definition of the lexicon). While this is an obvious property of spell-out at the upper interface of phonology (with morpho-syntax), it is counter-intuitive at its lower interface (with phonetics). The section also discusses why an arbitrary relationship of phonological and phonetic categories appears to be counter-intuitive, and how this issue dissolves in a substance-free approach to phonology.

While the first part of the article is concerned with the history and workings of the morpho-syntax – phonology interface,[2] the discussion of the second part focuses on more general properties of modular theory that go beyond language. It is argued that translation may turn out to always be lexical: all modules possess a lexicon that reduces variable inputs (distinct vocabularies) to a uniform output (the domain-specific vocabulary that is the input to modular computation). Relating to a thread of the Faculty of Language blog, objections against the generalization of lexical translation are discussed in section 7. In conclusion, it is shown that lexical translation not only qualifies for intermodular communication (within the cognitive system), but also for the association of real-world items with cognitive categories (e.g. wave lengths with colors).

## 1 THE STANDARD: TWO-CHANNEL TRANSLATION

The input to phonological computation is made of two types of pieces: morphemic and non-morphemic. In the string *#variable*, the item *variable* is a morpheme (or a sequence of morphemes), but # is not. The beginning of the word which traditionally appears as #

2 The first part of the article builds on a chapter of Scheer (2012: §160). Making the translation of both morphemic and boundary information lexical is an original idea by Michal Starke.

in phonology and may be relevant for phonological computation is not a morpheme because it does not have any meaning and is absent from the lexicon. Hash-marks represent information that is not stored but rather created online in the course of morpho-syntactic computation. That is, whether *variable* is preceded by # depends on its derivational history in morpho-syntax: in case it is preceded by a prefix as in *#in-variable*, the v- is not word-initial.

Let us thus refer to those pieces that appear in phonological representations and are of lexical origin as morphemic information, as opposed to boundary information which originates in morpho-syntactic computation. For the discussion below theory-specific properties of boundary information are irrelevant: hash-marks, units of the Prosodic Hierarchy, the initial CV or other carriers of morpho-syntactic information all represent boundary information.

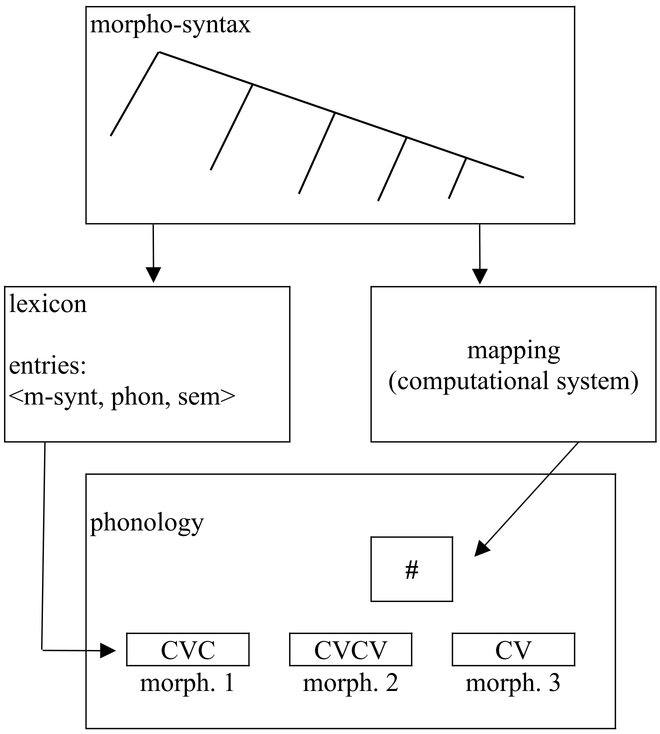
In production, the result of morpho-syntactic computation (the tree) is translated into phonological units through Vocabulary (or Lexical) Insertion. It is undisputed that those pieces of the tree that end up being represented by morphemic information in phonology are converted into morphemes through a lexical access. That is, the lexicon is a list of items stored in long term memory which matches phonological units with morpho-syntactic and semantic properties: something like <</kæt/>$_{\text{phon}}$, <animate, count etc.>$_{\text{morpho-synt.}}$, <beast of prey with claws>$_{\text{sem}}$> would be the lexical entry for *cat*.

Morphemic information in the phonology thus comes into being through a non-computational mechanism: Vocabulary Insertion matches portions of the freshly built tree with candidate lexical entries. There is no computational action that transforms morpho-syntactic into phonological units: the match is done in each lexical entry by a list-type conversion (look-up table).

By contrast in all interface theories since SPE the conversion of boundary information is computational in kind. That is, a computational action (often called mapping) transforms some portion of the morpho-syntactic structure into phonologically interpretable units. There is no lexicon, no storage and no list involved. Section 2 reviews in greater detail how computational translation of boundary information is implemented in various theories.[3]

The current situation in all interface theories may thus be described as two-channel translation: two distinct mechanisms convert morpho-syntactic into phonological vocabulary, one computational (boundary information), the other non-computational (list-type, morphemic information). This is depicted under (1).

(1) two-channel translation




3 The present article does not ambition to provide a full history of interface thinking. For the period since Trubetzkoy, this is the purpose of Scheer (2011). Rather, the focus is on generative and hence modular approaches to the interface, with a short prequel regarding structuralism in section 2.1.

# 2 THE TRANSLATION OF BOUNDARY INFORMATION IS COMPUTATIONAL IN ALL THEORIES SINCE SPE

## 2.1 STRUCTURALISM

There was boundary information in structuralist phonological representations, but it had to be incognito, pretending it was truly phonological and did not carry any morpho-syntactic information. This is because (in American structuralism) linguistic representations were supposed to be construed strictly bottom-up (discovery procedure), starting with a phonetic description from which phonological representations (the string of phonemes) was abstracted, which in turn was the input to the construction of morphological, finally of syntactic structure. Hence morpho-syntax was held not to influence phonology at all, but practitioners knew of course that this cannot be the case. The compromise were so-called juncture phonemes, for example # or +, which did import morpho-syntactic information into phonology but pretended to be regular phonemes, i.e. with free distribution (sic, therefore juncture phonemes could occur in the midst of morphemes) and a phonetic correlate (e.g. "either pause or nothing").

The structuralist situation is of course more intricate and manifold than this summary: the description in Scheer (2011: §59) goes into greater detail. The only thing that is relevant for the present purpose is the fact that there is no translation in the first place: the idea that morpho-syntactic structure is converted into phonological units by an explicit translational mechanism (for either morphemic or boundary information) was entirely foreign to structuralist thinking.

## 2.2 SPE

In SPE a fixed algorithm that works on the grounds of morpho-syntactic criteria alone converts boundary information into hash-marks: # is inserted into the linear string at the beginning and at the end of each major category (i.e. nouns, verbs and adjectives), and also on each side of higher constituents that dominate major categories, i.e. NPs, VPs and so forth (Chomsky & Halle 1968: 12f, 366ff).

In cases where (translated) syntactic surface structure does not qualify as the input for phonological interpretation, it needs to be modified before phonology can make use of it. This is what is called the Readjustment Component in SPE (Chomsky & Halle 1968: 9f, 371f) and what will later be known as non-isomorphism in Prosodic Phonology (see Scheer 2011: §§416ff). Since SPE the iconic example used in all subsequent literature is the cat-rat-cheese sentence (*This is the cat that caught the rat that stole the cheese*). The computational translation in SPE is described in greater detail in Scheer (2012: §§102f).

## 2.3 PROSODIC PHONOLOGY

### 2.3.1. Rule-based mapping

Prosodic Phonology operates the autosegmentalisation of linear hash-marks in the context of the general autosegmental movement of the 80s (Selkirk 1981 [1978], 1986, Nespor & Vogel 1986). Since computation at the time was rule-based, the translation of morpho-syntactic structure into prosodic arborescence (prosodic word, prosodic phrase etc.) was done by so-called mapping rules, which were running in a specific translational device that was located in modular no-man's land, i.e. distinct from both morpho-syntax and phonology (Selkirk 1984: 410f, Nespor & Vogel 1986: 302, see the overview in Scheer 2011: §§380f, 2012: §§85f). This initial setup of Prosodic Phonology faithfully instantiates Jackendoff's interface processors (discussed in section 4 below): a computational device that translates between two modules, is independent of both but able to process either vocabulary.

### 2.3.2. Constraint-based mapping

In the constraint-based environment of the 90s, mapping rules were replaced by constraint-based mapping (Selkirk 1996, see the overview in Scheer 2011: §457, 2012:

§88). On this take, the Prosodic Hierarchy is a device that represents the (mis)match of morpho-syntactic constituency and phonologically relevant chunks of the linear string. The operation that relates both, mapping, may thus be described as the (mis)alignment of the edges of morpho-syntactic constituents with the edges of phonologically relevant portions of the linear string (prosodic constituents in this approach). This perspective on mapping was introduced by Selkirk (1984: 52ff), who builds prosodic structure (the metrical grid in her case) by alignment rules (e.g. Basic Beat Rules, Demi-Beat Alignment DBA, see Scheer 2011: §426). Constituent margins then become the center of interest in Selkirk's (1986) edge-based mapping (Scheer 2011: §386). Finally, McCarthy & Prince (1993) generalize alignment to more empirical situations and make it the central tool of interface management with morpho-syntax in OT (see Itô & Mester 1999, McCarthy & Prince 2001: vii, Peperkamp 1995: 227ff for a historical overview).

In canonical OT, ALIGN is a constraint family with a uniform template: the left or right edge of a given unit coincides with the left or right edge of another unit. The units in question may be phonological, morphological or syntactic, and both units involved in an alignment constraint may belong to the same or different domains (phonology, morphology, syntax, see for example Yip 1998).

This move relocates mapping from a true interface position that stands outside of the two computational systems that are related into one of the systems at hand, phonology: alignment constraints are regular phonological constraints that are interspersed with other constraints in the same grammar (constraint hierarchy).

# 3 MODULARITY AND TRANSLATION

## 3.1 DOMAIN SPECIFICITY

Domain specificity is a heart piece of modularity. Coming from 19th century phrenology, the idea of modularity as exposed by Fodor (1983) is that the mind/brain is made of a number of distinct computational systems that are specialized in particular tasks, rather than of one single and all-purpose system. Each module is thus distinct from each other and only competent for a specific domain: this is called domain specificity. It follows that it is incompetent for, or insensitive to anything that is done in neighbouring modules: it cannot parse or understand information that is foreign of its own domain.

Thus Coltheart (1999: 118) says that "a cognitive system is domain-specific if it only responds to stimuli of a particular class". A longer summary by Peter Carruthers appears under (2).

(2) "According to Fodor (1983) modules are domain-specific processing systems of the mind. Like most others who have written about modularity since, he understands this to mean that a module will be restricted in the kinds of content that it takes as input. It is restricted to those contents that constitute its *domain*, indeed. So the visual system is restricted to visual inputs; the auditory system is restricted to auditory inputs; and so on." Carruthers (2006: 3f, emphasis in original)

The domain-specific character of modules is thus determined by the kind of input that they can parse and compute: this input is restricted in such a way that it is specific to their domain. This is why it is often said that modules work on their own proprietary vocabulary (or alphabet). Hence, "[t]o say that a system is domain specific is to say that it only receives inputs of a particular sort, concerning a certain kind of subject matter" (Carruthers 2006: 7).

Even though a variety of approaches to modularity is entertained, domain-specificity is common to all versions.[4] It excludes a situation where a given module is computing

4 Fodor (2000: 58-64) and Carruthers (2006: 3-12) discuss different versions and uses of domain specificity. They are essentially irrelevant for the argument made in the present article. Distinctions include how the domain-specific restrictions are defined (through the input to the computational system, or through its function), whether there is a difference between different types of inputs (from storage vs. from other modules) and whether domain specificity is also a property of modules in massive modularity, i.e. when what Fodor (1983) calls central systems are also considered to be modules (Sperber 2001). The latter issue is orthogonal to the purpose of the present article, where only classical lower cognitive functions (language and its subsystems, the five senses) are considered.

items that do not belong to its domain. Thus Jackendoff (1997: 87): "'Mixed' representation[s] should be impossible. Rather, phonological, syntactic and conceptual representations should be strictly segregated, but coordinated through correspondence rules that constitute the interfaces."

In this quote, Jackendoff also mentions an immediate and necessary consequence of domain specificity: since modules can only parse a restricted set of input items that corresponds to their domain, they are unable to directly communicate with their neighbours and hence to do that an interface is needed that translates among them (what Jackendoff at that time calls correspondence rules).

This is not to say, of course, that there is a ready-to-use cartography of modules where one can look up the kind of computational systems that are around. Identifying them is not a trivial task and usually, like elsewhere in science, initial coarse-grained modules are found to have internal structure as inquiry proceeds. Hence Fodor's (1983: 47) first approximation was a set of 6 modules: the five senses plus language. But he pursues saying that "I imagine that within (and, quite possibly, across) the traditional modes, there are highly specialized computational mechanisms in the business of generating hypotheses about the distal sources of proximal stimulations. […] Candidates might include, in the case of vision, mechanisms for color perception, for the analysis of shape, and for the analysis of three-dimensional spatial relations" (Fodor 1983: 47). Indeed vision is now thought to fall into a number of modular subsystems that are responsible for shape, colour, movement, face recognition or contrast (Marr 1982 and following, see Stevens 2012 for an overview).

In the same way, today language is thought of being decomposed into a number of distinct modules: morpho-syntax (domain-specific vocabulary: person, number, gender and so on) and phonology (domain-specific vocabulary: occlusion, labial, voice and so on) for sure, depending on authors also semantics, phonetics or pragmatics (e.g. Jack-

Further references where domain specificity is discussed (and which are explicit about it restricting the access to a certain kind of information) include Collins (2017: 224) and Jackendoff (1997: 41, 2002: 218-227).

endoff 2002, Chomsky 2000: 118). How to identify modules and the right granularity of their domain is discussed e.g. by Coltheart (1999: 117-120).

Hence the proper identification of a given module and its specific purview (domain specificity) is an empirical question and may vary over time. But domain specificity itself is invariable: whatever its domain, a module cannot parse or compute information that lies outside of this domain.

## 3.2 CONSEQUENCES FOR TRANSLATION

As was mentioned in section 3.1, domain specificity is the *raison d'être* of interfaces: the domain-specific vocabulary of different modules is mutually unintelligible and thus needs to be translated into the language of the receiving module before communication can occur.

Domain specificity rules out two approaches to translation: constraint-based mapping (section 2.3.2) and more generally computational translation. The latter is refuted because it requires a computational system that converts one type of vocabulary into another and hence is able to parse two distinct vocabularies belonging to two distinct domains, its input and its output. Recall from section 2 that all theories to date concerned with the translation of boundary information are computational in kind and therefore incompatible with modular standards.

Constraint-based mapping is computational in kind and therefore does not qualify anyway. But on top of that it places translation *in* a particular module, phonology, instead of managing it in modular no-man's land (which is where the earlier rule-based incarnation of mapping occurred, following modular standards). Doing translation in the phonology is incompatible with modularity because phonology is unable to understand or parse vocabulary from foreign domains. Alignment constraints therefore do things that are impossible: they read morpho-syntactic vocabulary while being located in the phonology.

The following section describes Ray Jackendoff's model of modularity which is entirely based on computational translation.

## 4 JACKENDOFF'S MODEL OF TRANSLATION (IN LANGUAGE AND ELSEWHERE)

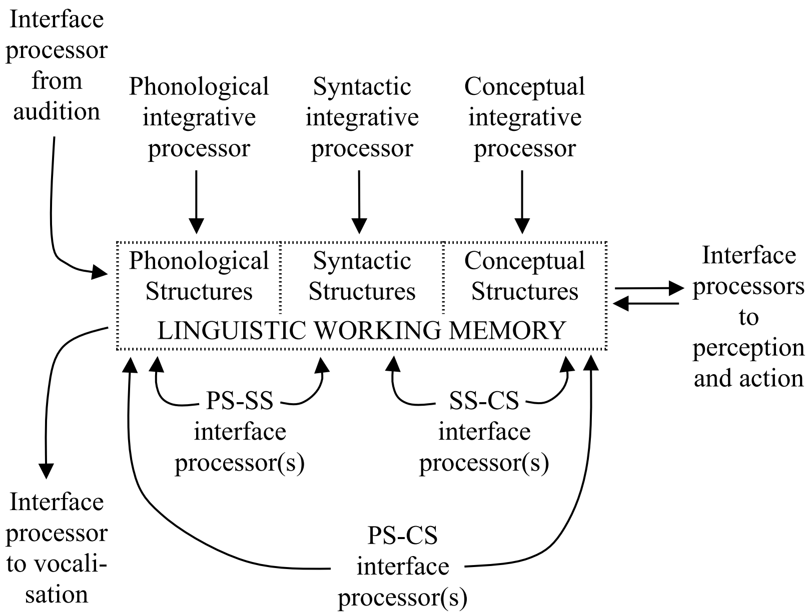
### 4.1 TRANSLATION IS DONE BY COMPUTATIONAL SYSTEMS WITH MODULAR STATUS

Ray Jackendoff's (1992, 1994, 1997, 2002) model of modularity was first known as Representational Modularity, later on as Structure-Constrained Modularity. In his view intermodular communication is computational in kind: "it is not like sending a signal down a wire or a liquid down a pipe. It is, rather, a computation in its own right, just the kind of computation that an interface processor performs" (Jackendoff 2002: 223). In Jackendoff's system, interface modules relate two regular modules by accessing their content simultaneously and transferring information from one to the other. Jackendoff (2002: 223, note 19) is explicit on the modular status of the computational system that carries out translation (see also the quotes in sections 4.2 and 4.3).

Jackendoff thus promotes a general modular architecture of the mind where three types of modules (which he calls processors) are active: inferential processors (Fodor's central systems, i.e. which construct inferences and judgments), integrative processors (Fodor's domain-specific modules, e.g. color recognition, paucal counting, phonology, syntax etc.) and interface processors. Integrative processors are related by interface processors, which also assure the communication with Fodorian central systems (Jackendoff's inferential processors).

The modular structure that Jackendoff (2002: 199) proposes for language in this general environment is shown under (3) below.

(3) modular structure of language according to Jackendoff (reproduction of Jackendoff's 2002: 199 diagram)




## 4.2 JACKENDOFF'S COMPUTATIONAL TRANSLATION IS ALL-POWERFUL AND UNCONSTRAINED

Jackendoff considers that translation among modules is entirely unconstrained: it can do anything that needs to be done for the information flow to work, and must not be limited in any way. His position in this respect has not varied since his earliest work: interface processors appear as translation rules in Jackendoff (1987), as correspondence rules in Jackendoff (1997).

Jackendoff explicitly defends all-powerful translation against the critique of overgeneration, i.e. the fact that unconstrained transmission of information allows for the description of existing as much as of non-existing interface activity.

(4) "[C]orrespondence rules are conceptually necessary in order to mediate between phonology, syntax, and meaning. It is an unwarranted assumption that they are to be minimised and that all expressive power lies in the generative components. […] In other words, correspondence rules, like syntactic and phonological rules, must be constrained so as to be learnable. Thus their presence in the architecture does not change the basic nature of the theoretical enterprise." Jackendoff (1997: 40)

The quote also shows that Jackendoff conceives of correspondence rules in the same way as of phonological or syntactic processes: translation is modular computation.

### 4.3 BI-DOMAIN SPECIFICITY

A direct consequence of the computational and hence modular status of translation is what Jackendoff calls *bi-domain specificity* (Jackendoff 2002: 220ff). In his 1997 book, he provides the following description of correspondence rules: "correspondence rules perform complex negotiations between two partly incompatible spaces of distinctions, in which only certain parts of each are 'visible' to the other" (Jackendoff 1997: 221). Hence interface processors have only access to a subset of the structure that is present in either of the two modules that are related.

In this sense, interface processors define their own domain of competence, which is composed of a subset of the structure of each module that they relate. Hence Jackendoff's term bi-domain specificity: "[e]ach module is strictly domain-specific in Fodor's sense: integrative and inferential processors deal with only one level of structure each; interface processors deal with two (we might therefore want to call them 'bi-domain-specific')" (Jackendoff 2002: 2002).

Bi-domain specificity is incompatible with modularity (see section 3.2). It merely puts a word on a contradiction in terms: a module cannot be bi-domain specific since the whole idea of modularity is that computational systems are not all-purpose but rather restricted to work on a specific domain. Hence they cannot take into account inputs that belong to another domain.

Bi-domain specificity also defies the purpose of interfaces (translators): if modules were able to process two different types of vocabulary, there would be no need for specific translation devices in the first place. The existence of interface processors in Jackendoff's system under (3) would be useless since the regular (integrative) processors could do the same job, were they bi-domain specific. There is no rationale in Jackendoff's writings, and no explanation, why only interface modules can be bi-domain specific: if modules can compute inputs from more than one domain, why is it that content-processing modules do not have this faculty?

Finally, in a situation where as in Jackendoff's model all translation is computational in kind it is unclear what the status is of the uncontroversially list-based (i.e. non-computational) translation between morpho-syntax and phonology (Vocabulary Insertion: past tense ↔ -ed, *not* past tense → -ed).

## 5 INTERMODULAR COMMUNICATION GOES THROUGH A LEXICAL ACCESS

If computational translation does not qualify in a modular environment, the only alternative available to date appears to be a lexical access (but see section 7.5 for discussion of the approach relying on similarity calculus). Lexical translation is well documented in its instantiation as Vocabulary Insertion at the interface of morpho-syntax with phonology and respects modular standards. The present section explores the workings of lexical translation by looking at more general properties of intermodular communication that Ray Jackendoff has worked out.

### 5.1 MODULES RECEIVE VARIABLE INPUTS, BUT PRODUCE A UNIFORM OUTPUT

Modules may draw on information that comes from a range of other modules (many-to-one); conversely, the output of a given module may be used as the input to a range of other modules (one-to-many). Jackendoff (2002: 223f) reviews a number of relevant cases.

Audition for example is an information provider for a number of quite different modules: sound is processed by all-purpose audition (e.g. the perception of sound that is produced by animals), voice recognition (the identification of humans according to their voice), auditory affect perception (emotion detector) and the perception of linguistically relevant phonetic material. How exactly the phonetic signal that hits the sensory system is chopped into the pieces that are relevant for the various clients mentioned is unclear to date, but it is a fact that relevant information finds its way to each computational system.

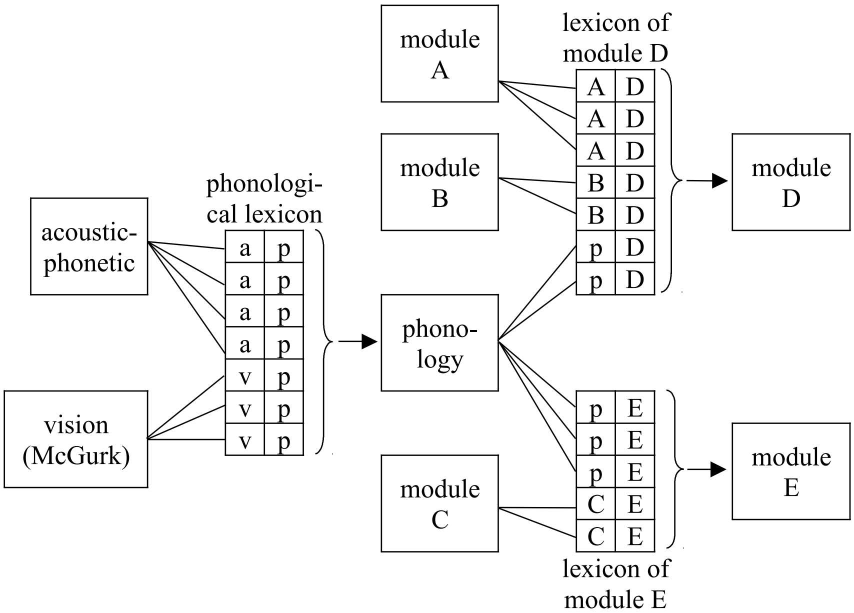
The reverse situation is encountered when the same module receives input from different sources. In perception for example, phonology is fed at least by acoustic-phonetic and visual information. The latter is documented by the McGurk effect (McGurk & MacDonald 1976, Ingleby & Azra 2003): when exposed to auditory and visual information that simultaneously provide conflicting information, subjects perceive something that is absent from the sensory input. Exposed to auditory [ba] and visual "[ga]" (in a video where the person is filmed pronouncing [ga], but with a synchronized audio track that plays [ba]), they perceive [da] (so-called McGurk fusion). The functional rationale for the phonological system being wired to receive visual input is lip reading, which enhances perception in noise-impaired environments and is (unconsciously) practiced by all humans. Calvert & Cambell (2003) have found that the circuitry of visual stimuli that reach grammatical processing is different from the one used for auditory stimuli, but processed by the auditory cortex.

## 5.2 THE LEXICON REDUCES A VARIABLE INPUT TO A UNIFORM OUTPUT

Modules are thus able to compute information from a variety of sources whereby each sending module uses a different vocabulary. This means that a (lexicon-based) translation process converting all incoming data into the vocabulary of the receiving module must be operating before the receiving module can process anything. That is, on their input side modules have a lexicon whose own input comes in a number of distinct vocabularies, but whose output are always units of the to-be-computed vocabulary.

Table (5) below shows how a modular network communicates through lexical access along the lines discussed (note that the language bit concerns perception). Lexical entries are pairs of arbitrarily associated items which belong to two different domains. The input to the phonological lexicon falls into items of the acoustic-phonetic ("a") and the visual vocabulary ("v"). These units are associated to pieces of the phonological vocabulary ("p"): a ↔ p and v ↔ p.

(5) intermodular communication through lexical access




An example from perception where a module receives input from multiple vocabularies was mentioned in the previous section and is depicted under (5): on top of acoustic information, phonology also receives visual input (McGurk effect). In production, phonetics receives the output of phonology, but also input from pragmatics and emotion processing: the phonetic signal conveys information from (at least) all these domains.

# 6 CONSEQUENCES OF LEXICAL TRANSLATION

## 6.1 ARBITRARINESS

An obvious and necessary property of lexical translation is the arbitrariness of the relationship between the items that are associated. Hence in a spell-out instruction x ↔ α that is stored in long-term memory (as a piece of the spell-out lexicon) where x belonging to the "Latin" vocabulary is related to α of the "Greek" vocabulary, the relationship between both items is necessarily arbitrary: there is no way in which x and α could even be compared, either by the analyst or by the cognitive system – they are incommensurable because they belong to different, mutually unintelligible vocabularies. This is just like in a real language dictionary, say, English – Czech, which has an entry table ↔ stůl: there is no reason for the Czech equivalent to be *stůl* rather than, say, *kad*, *rtim* or *trob*. The relationship cannot be predicted and any two items of the respective vocabularies could in principle be associated.

Arbitrariness is an obvious and undisputed property of Vocabulary Insertion, i.e. the translation of morphemic information when spell-out transforms morpho-syntactic structure into phonological units. Arbitrariness upon Vocabulary insertion is never argued for because it goes without saying: no argument needs to be made. A morpho-syntactic structure that describes, say, past tense of a weak verb in English is realized as -*ed* because there is a lexical entry stored in long-term memory that specifies this equivalence: past tense [weak verbs] ↔ -*ed*. There is no reason why -*ed*, rather than, say, -*s*, -*et* or -*a* realizes past tense in English.

Note that arbitrariness is a necessary property of lexical translation, but may also be produced by some versions of computational translation: anything in the input vocabulary can in principle be turned into anything in the output vocabulary. Jackendoff (1997: 40) is explicit on all-powerful computational translation: "correspondence rules are conceptually necessary in order to mediate between phonology, syntax, and meaning. It is an unwarranted assumption that they are to be minimised and that all expressive power lies in the generative components."

The difference between lexical and computational translation is that arbitrariness is a necessary and immutable property of the former, but only a possibility of the latter.

## 6.2 THE OUTPUT OF TRANSLATION MUST BE A GOOD LEXICAL ENTRY

Unlike computational translation, lexical translation constrains possible associations of items belonging to two different vocabularies by a condition on the output: the result of translation must be a good lexical entry, i.e. belong to the domain-specific vocabulary of the receiving module.[5]

As a consequence, diacritics do not qualify. In all phonological theories to date, carriers of morpho-syntactic information in phonology are diacritics: juncture phonemes, hash-marks or items of the prosodic constituency (omegas ω, phis φ, etc.) and the like are all arbitrarily chosen and interchangeable units that do not belong to the phonological vocabulary (labial, occlusion, voice, etc.) and are therefore phonologically meaningless. In current and past theories that use diacritics, these are not stored in the lexicon

5 The definition of what a good lexical entry is, i.e. of what may be stored and what may not (or whether there is any such restriction at all), is a lively discussed issue in the literature. It is useful to keep the discussion about the interface as theory-neutral as possible in order to guarantee the independence of its conclusions. Below some approaches to what counts as a good lexical entry are mentioned. I personally favour the classical position reported.
The classical position formalized as Morpheme Structure Constraints (MSC) in SPE (Chomsky & Halle 1968: 171, 382) holds that a good lexical entry is one that is well-formed according to the standards imposed by the phonology of the language at hand. For example, in a language like English where only clusters of rising sonority can begin a morpheme, lexical items with initial #RT clusters are prohibited, i.e. will never be lexicalized. Relevant literature includes Rasin & Katzir (2015), Rasin (Ms.) and Gouskova & Becker (2016), the latter authors talking about a Gatekeeper Grammar.
So-called Lexicon Optimization embodies the same idea. Bermúdez-Otero (2003: 29) provides the following formulation (after Hale 1973: 420): prefer inputs that are well-formed outputs. Relevant literature includes Prince & Smolensky (1993: §9.3), Yip (1996), Bermúdez-Otero (1999: 124) and Inkelas (1995).
In classical versions of OT, the restriction of lexical entries to well-formed items (or any other restriction for that matter) conflicts with Richness of the Base (Prince & Smolensky 2004 [1993]: 191, McCarthy 1998). Relevant discussion can be found in the volume on Freedom of Analysis edited by Blaho et al. (2007), as well as in BlahoRasin (2018: 93-152), Rasin & Katzir (2015) and Vaux (2005).

but rather the result of computational translation (see section 2). This is consistent since only items of the domain-specific vocabulary of a module are storable. If translation is lexical also for boundary information, diacritics are thus disqualified.

Aside from being unable to be stored, diacritics also cannot be parsed by phonological computation: phonology can only process items of its own vocabulary. The diacritic issue is discussed at greater length in Scheer (2008, 2011: §§402, 687, 2012: §§66, 93). The alternative is a non-diacritic interface theory (Direct Interface, Scheer 2012) where the output of translation can only be made of items that belong to the phonological vocabulary, i.e. items which exist in the phonology in absence of morpho-syntactic conditioning.[6]

### 6.3 ARBITRARINESS ALSO AT THE LOWER INTERFACE OF PHONOLOGY (WITH PHONETICS)

If intermodular communication goes through a lexical access, then interfaces other than the one between morpho-syntax and phonology must instantiate the same architectural properties. Hence the interface of phonology with phonetics must be organized in terms of a spell-out operation whose input (phonological categories) entertains an arbitrary relationship with its output (phonetic categories).[7] This is referred to as phonetic arbitrariness below.

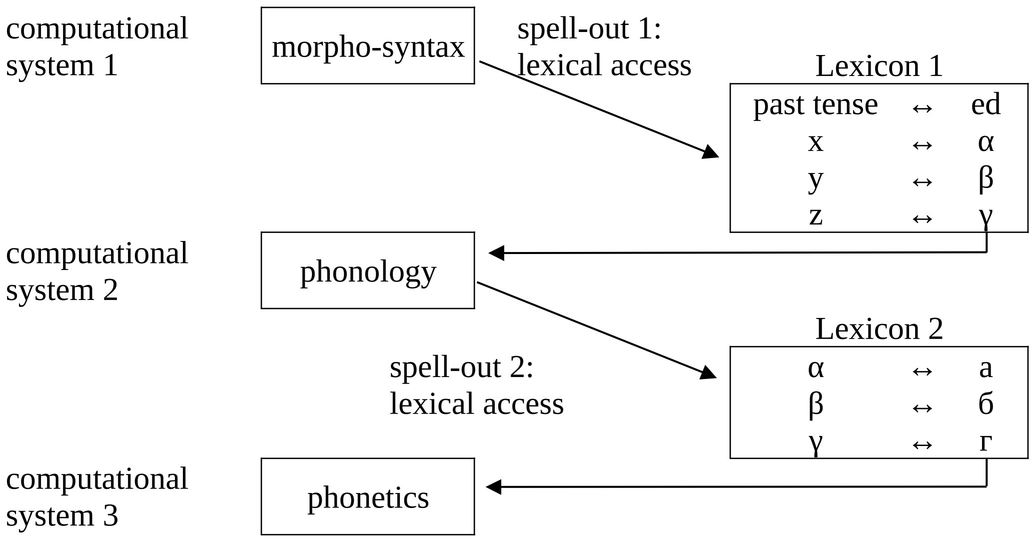
The resulting architecture is shown under (6) below: the lower interface of phonology (with phonetics) has the same workings as its upper interface (with morpho-syn-

6 This is the definition of "diacritic": an item that is foreign in a given set of units. The hash mark was said to be a diacritic because it was added to vocabulary items that were segments. In modular terms, thus, a diacritic is an item that does not belong to the input vocabulary of a given module. Hash marks, omegas etc. are thus diacritics in phonology: they are absent from phonological representations or processes when phonology is by itself, i.e. in absence of morpho-syntactic conditioning. Their only purpose is to introduce morpho-syntactic information. In the alternative mentioned, carriers of morpho-syntactic information in phonology are true members of the domain-specific phonological vocabulary that exist independently of interface issues: syllabic space (x-slots, moras, CV units depending on the theory favoured).

7 Of course this is true only if phonology and phonetics are distinct modules. The alternative where they are scrambled in the same computational system is entertained e.g. by Steriade (1999), Flemming (2004), Hayes et al. (2004).

tax). Three distinct computational systems, each processing a specific vocabulary distinct from the two others, communicate through a translational device (spell-out) that is identical: an input item in some vocabulary is converted into an output item in another vocabulary through a lexical access.

(6) fragment of (production) grammar involving phonology




Arbitrariness at the phonology-phonetics interface is counter-intuitive because unlike at the upper interface our experience is that the relationship is 99% faithful: what is labial in phonology is also labial in phonetics. There are distortions, though, which show that translation may be non-faithful. For example, the segment that is usually transcribed as r occurs as the second member of branching onsets in a number of languages, thereby identifying as a phonological liquid – but it is pronounced as an obstruent. This is the case for example in Polish where the liquid at hand appears as [ʃ,ʒ], in Brazilian Portuguese where it is [h] or in French and German where it is pronounced [ʁ,χ] (Chabot 2019).

Scheer (2014) discusses two reasons why 99% of translational relations at the lower interface of phonology (with phonetics) are faithful (while faithfulness cannot even be expressed at the upper interface with morpho-syntax: the correlation of "past tense" with -*ed* is not any more or less faithful than if it were associated to -*a*, -*ub* etc.). One is about the ontology of morpho-syntax (only grammar-internal, no real-world categories are involved), as opposed to phonetics (concerned with real-world categories that lie outside of the cognitive system). Section 7.4 below expands on this difference. The other reason is the diachronic origin of non-faithful relationships at the lower interface: rules and phonology – phonetic mappings are not born crazy, they become crazy through aging (Bach & Harms 1972, more on this in section 7.5).

A consequence of the arbitrary mapping of phonological onto phonetic categories is that the labels which are traditionally used by phonologists lose their meaning. If in a given language the phonological item [labial] is realized as phonetic palatality, the question arises why anybody would call the phonological object in question "[labial]". In the cases mentioned where a mismatch is observed, typically analysts call on the phonological identity when it comes to naming the object: something that behaves phonologically as a rhotic because it can be the second member of a branching onset will be called a rhotic, or r, even in case it is pronounced [ʃ,ʒ], [h] or [ʁ,χ].

Substance-free phonology has taken the step of completely eliminating phonetic content, and hence phonetic labels, from phonology. Phonological items thus identify as colourless α,β,γ etc. shown under (6) whose only purpose is to support contrast and to allow for computation (phonologically active classes). Phonetic categories only come in through the spell-out relation (e.g. α ↔ labial).[8]

Cutting across theories, voices which have reached the conclusion that phonological items are substance-free (alphas, betas etc.) and entertain an arbitrary relationship with phonetic categories include the BiPhon model (Boersma 1998: 461ff, Boersma 2011, Boersma & Hamann 2008: 263, Hamann 2011, 2014), Mielke (2008), Carvalho (2002),

8 In this perspective a legitimate question is in which way the vocabulary of different modules is distinct: what is the difference between a "phonological alpha" and a "semantic alpha", or an alpha coding some cognitive colour category for that matter? This question is orthogonal to the present argument; it is addressed in Scheer (forth).

Odden (2006, 2019), Blaho (2008), Samuels (2012), Iosad (2012: 6ff, 2017), Scheer (2014), Chabot (2019), Dresher (2014, 2018) (see the summary in Scheer 2019).[9]

## 7 DISCUSSION ON THE FACULTY OF LANGUAGE BLOG

An exchange on Norbert Hornstein's and Bill Idsardi's Faculty of Language blog dated April 2019 is relevant to the discussion above. The sections below look at some issues raised.

### 7.1 BI-DOMAIN SPECIFICITY ALSO FOR NON-INTERFACE MODULES?

In his post Dueling Fodor interpretations (permalink) (3 April 2019), Bill Idsardi raises the question exactly how specific domain-specific Fodorian modules are: could the granularity of their specificity be larger, embracing not just one but several domains? Note that in Jackendoff's system bi-domain specificity is only a property of interface modules (his interface processors), regular content-computing modules (his integrative processors) being specific to only one type of vocabulary. Idsardi thus considers extending Jackendoff's bi-domain specificity to regular modules that do not carry out translation.

Idsardi quotes Fodor's (1983: 47) reflection on the McGurk effect, leading Fodor to admit that modules can operate "within (and, quite possibly, across) the traditional modes" (what he calls modes are the five human senses). Crucial here is "across", which suggests that modules can process vocabulary from more than one source. In a footnote explaining why he has included "across", Fodor calls on the McGurk effect which shows that the language module processes both auditory and visual information. Fodor

9 Hale & Reiss (2000, 2008), Hale & Kissock (2007: 84), Volenec & Reiss (2018: 253, 2019) also represent substance-free phonology, but do not endorse phonetic arbitrariness: they hold that the set of phonological primes as well as their mapping onto phonetic categories is universal and genetically coded (Scheer 2019 provides an overview of this option).

In the alternative view held by the authors quoted in the main text, spell-out relations are language-specific and hence acquired by the child. The relationship between phonetic and phonological items that they describe is thus arbitrary in principle, i.e. cannot be predicted for any given language (but of course once a speaker has acquired the spell-out relations of their language, they are fixed).

insists on the unity of the domain in which this processing occurs, though, which is neither auditory nor visual: "[i]t is of central importance to realize that the McGurk effect – though cross-modal – is itself domain specific – viz., specific to language" (Fodor 1983: 132). Hence Fodor sticks to a domain specificity that can process only one vocabulary: the one of "language".[10] This means that Fodor relies on regular intermodular translation which transforms the auditory and the visual vocabulary into the domain-specific "language" vocabulary, which is then processed by the language module. Whether this translation is computational or lexical in kind is irrelevant for the argument: what matters is that it exists. That is, modules do not directly process information provided from other modules – this information is always filtered by a translational device, the interface.

Given this situation there is no evidence in the location referred to by Idsardi that Fodor has envisioned a violation of domain specificity, i.e. the possibility that a given module processes information from several distinct domains. Quite to the contrary Fodor upholds that the language module only processes information pertaining to language, vision and audition input having been translated into language-type vocabulary beforehand. The literature reviewed in section 3.1 is also unanimous on domain-specificity as a necessary property of modules: the whole idea of modularity is that computation is not all-purpose but domain-specific. There is no modularity without domain-specificity.

Also, as was mentioned earlier, note that if modules could process more than one vocabulary, there would be no need for translation in the first place.

In his post Idsardi also mentions other cases of multisensory integration that are detected by brain-based evidence. These are thus other cases along the lines of the McGurk effect: the existence of multisensory input to a module is trivial and undisputed (see section 5). This does not mean that modules directly process distinct sensory vocabulary, though: different kinds of vocabulary are translated into the receiving mod-

10 See section 3.1 on the further decomposition of what Fodor thought of as the language module in a first approximation.

ule's vocabulary prior to being computed. There is no evidence that Fodor has ever departed from this position.

## 7.2 GALLISTEL & KING'S *INFINITUDE OF THE POSSIBLE* DOES NOT APPLY TO INTERFACES

In his post More on "arbitrary" (permalink) (29 March 2019), Bill Idsardi argues that lexical translation (look-up table in his terms) is a strange fellow that we do not want to generalize – it should be considered only as a last resort when everything else has failed. Idsardi says "spell-out via lookup table is literally the weakest possible architectural assumption about transduction." He motivates this statement with a quote from Gallistel & King (2010: xi) where the authors "make a critical distinction between procedures implemented by means of lookup tables and what we call compact procedures." Gallistel & King argue against look-up tables on the grounds of their computational inefficiency, as compared to a computational procedure:

(7) "the specification of the physical structure of a look-up table requires more information than will ever be extracted by the use of that table. By contrast, the information required to specify the structure of a mechanism that implements a compact procedure may be hundreds of orders of magnitude less than the information that can be extracted using that mechanism." Gallistel & King (2010: xi)

The high memory and accessing cost of look-up tables such as the linguistic lexicon that is used in Vocabulary Insertion thus makes lexical translation undesirable: the weakest possible solution.

A crucial piece of information that needs to be added, though, is that Gallistel & King only talk about modular computation (which may describe infinite sets of items): they are not concerned with interfaces (whose input and output is necessarily finite).

The point they make is about what they call the *infinitude of the possible*, i.e. the fact that an infinite set of items cannot be materialized by a list or a look-up table: it can only be described by a (mathematical) function. Examples discussed by Gallistel & King (2010: 51-53) include the multiplication function over the infinite set of integers (f*: $\mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$, the output of which cannot incarnate as a list since, like its input, it is infinite) and the prime number function (which, given a prime number, produces the next higher prime number $f_{next\_prime}$ (n) where n is an integer: nobody can compute or list the exhaustive output).

It is an obvious and trivial fact that an infinite set of items cannot be actually computed or written down on a piece of paper – but that it may be described by a mathematical function. Hence whenever infinity enters the game, look-up tables are not an option. The question then is in which way infinity is involved in cognitive and modular activity.

Chomsky's canonical description of language as a *discrete infinity* expresses the fact that the infinite number of well-formed sentences of a language is created by a finite and discrete set of basic building blocks that undergo computation. These building blocks are the items of the morpho-syntactic (input to morpho-syntactic computation) and the phonological (input to phonological computation) domain-specific vocabularies, which are necessarily finite since they are stored in long term memory. The conclusion following Gallistel & King is that the modular computation of morpho-syntax cannot be stated in terms of a look-up table but only as a function, since it produces infinity. Its input items are for sure listed in long-term memory in the form of a look-up table, though.

The next question is whether anything infinite does, will or could occur in intermodular communication. The answer, I think, is no. The input to modular computation is necessarily a finite set of stored lexical items. This is true for syntactic computation, but also for perceptual modules like vision, audition etc.: given domain specificity, all modular activity is based on a vocabulary that by definition cannot be infinite because it needs to be stored. Hence Gallistel and King's *infinitude of the possible* argument holds for modular computation where indeed functions are running that may describe an infinite set of outputs. But whatever the output of modular computation, it is made of the initial vocabulary items that were combined / rearranged by modular computation. Their set is the same as before the computation started, that is finite. Therefore when translational devices convert the output of one module into the relevant input vocabulary of another module, the set of items to be translated is always finite.

In conclusion, Gallistel & King's *infinitude of the possible* does not apply to intermodular communication, which only ever translates one finite set of items into another. Hence there is no conceptual superiority of computational over lexical translation ("weakest possible architectural assumption about transduction"). There is reason to believe that the reverse is actually the case: it was shown above that computational translation is incompatible with modularity because it violates domain specificity.

### 7.3 IS LEXICAL TRANSLATION THE ODD MAN OUT?

In his post More on "arbitrary" (permalink), Bill Idsardi also argues against the generalization of lexical translation on the grounds of the fact that as we know it from Vocabulary Insertion it has no obvious parallel elsewhere in the cognitive system. He quotes Jackendoff (1997: 107) on this: "[i]f we look at the rest of the brain, we do not immediately find anything with these same general properties. Thus the lexicon seems like a major evolutionary innovation, coming as if out of nowhere."

The (empirical) question is thus whether there is evidence beyond the morpho-syntax – phonology interface that lists of items are matched when two modules exchange information. Aside from the fact that interfaces necessarily map one set of domain-specific and finite vocabulary onto another (section 7.2), there is ample descriptive evidence for this exchange to imply finite lists whose members are matched.

In language production there is a list of phonemes, which is decomposed into a list of melodic primes (features), and these correspond to a list of phonetic categories (acoustic target values for instance in Boersma & Hamann's 2008 BiPhon model). That is, the items of a phonological and a phonetic list are matched.

In color perception, a gradient spectrum of wave lengths is discretized into a finite number of distinct colors that people perceive. The slicing of the real-world spectrum into cognitive categories (colors) may be more or less fine-grained according to culture, language (Sapir-Whorf), expertise (a tissue salesman will distinguish more colors, each with a specific word, than non-professionals) etc. (e.g. Grieve 1991, Choudhury 2014: 144-184, Whittle 2003, Webster 2003, Hansen et al. 2006, Winawer et al. 2007, Athanasopoulos 2009). But the number of cognitive categories is always finite, and they always correspond to a real-world item, i.e. a range of wave lengths. The same goes for all other senses: odor, taste, sense of touch, audition. In all cases a list bearing a finite set of items is involved on the cognitive side (cognitive categories).

Given this record, listed items pertaining to one vocabulary that are matched either with listed items of another vocabulary or with portions of a real-world continuum, appear to be ubiquitous in the cognitive system. The question is thus rather whether there is any evidence that there is intermodular communication which is *not* lexical, i.e. does not match items of two lists of distinct vocabulary.

What would a computational translation of, say, the wave length-to-color mapping look like? A computation could transform wave length X into color A, wave length Y into color B etc. If it were the case that the nanometer value of color boundaries can be calculated, an algorithm could compute these boundaries. But even then whenever a boundary is detected some device will have to match this boundary with an actual (cognitive) color. Alternatively, computational translation could be done by devising a distinct computational instruction for each match of items on the two lists: wave length X → color A, wave length Y → color B etc.

Computational translation thus appears to be just a different way of referring to the fact that in intermodular communication items of two lists of distinct vocabularies are matched. That is, computational translation is but a notational variant of the intrinsically lexical character of translation.

### 7.4 CROSSING THE REAL-WORLD BOUNDARY OR NOT

Intermodular communication occurs between two modules, i.e. two computational systems of the mind. Morpho-syntax and phonology are two cases in point. But what about, say, phonology and phonetics? Or any of the five senses and their cognitivized correspondents? In these cases the real-world boundary is crossed: a gradient signal that lies outside of the cognitive system is boxed into discrete cognitive categories. The association of wave length (real world) and color (cognitive category) is roughly as under (8) below, with the kind of factors mentioned in section 7.3 that modulate the picture: culture, language, expertise etc.

(8) correspondence wave length – colors
(Choudhury 2014: 7)

| wave length (nm) [real world] | color associated [cognitive category] |
|---|---|
| 380-450 | violet |
| 450-480 | blue |
| 480-510 | blue-green |
| 510-550 | green |
| 550-570 | yellow-green |
| 570-590 | yellow |
| 590-630 | orange |
| 630-750 | red |

The correspondence is certainly of the lexical kind, i.e. relating items of two different sets of vocabulary in an arbitrary fashion. As was mentioned, no matter into how many different colors the wave length continuum will be divided (experts may have more categories than non-experts), their number is finite.

But the real-world side of the table is certainly not a computational system, nor is it cognitive in kind. Hence it can hardly be said that the lexical translation under (8) is an instance of intermodular communication.

The conclusion is that lexical translation seems to be a good candidate for all purposes: it relates modules within the cognitive system as much as units of the cognitive system with all sorts of real-world signals that come in through the five senses.

## 7.5 SIMILARITY CALCULUS

The difference between intra-cognitive translation and cognitive–real world association is the reason why there is an intuitive similarity calculus in the latter, but not in the former case (Scheer 2014: 268f).

It is expected that the phonological prime [labial] (cognitive category) will also be phonetically realized as labiality (real world).[11] Otherwise analysts talk about a "mismatch" or a "non-faithful" mapping (see section 6.3). The overwhelming majority of phonology – phonetics associations is faithful and hence produces the (phony) impression that there is some necessity or inclination for faithful mapping, which is expected and regular. Mismatches exist, but are held to be awkward and odd. This is what Bill Idsardi (in his post Dueling Fodor interpretations permalink) calls partial veridicality: the similarity calculus between phonetic and phonological items produces a faithful (veridical) match most of the times, but sometimes the correspondence is non-faithful (non-veridical).

As was mentioned in section 6.3, by contrast lexical translation predicts that the association of phonological and phonetic categories is arbitrary: hence as far as the cognitive system is concerned, odd non-faithful mappings are just as regular and expected as faithful correspondences. They are less frequent because all phonological processes are regular and phonetically faithful at birth since they result from the phonologization of a phonetic precursor. Non-faithfulness comes about when phonological processes age and

11 Where exactly the red line runs between cognitive computational systems and the real world is a question open to debate, but orthogonal to the present discussion. At some point a cognitive phonological category will end up as an articulatory and acoustic item, whatever the sequence of computational systems is that intervene. The phonetic literature today is fairly consensual regarding the existence of a language-specific phonetics that imposes well-formedness conditions (and hence is cognitive in kind and acquired by the infant), distinct from universal phonetics (Keating 1985, Cohn 1998, Kingston 2019: 389). The former may be acoustic, the latter articulatory in kind.

are telescoped with other processes (Bach & Harms 1972), but on most occasions processes run through the phonological life cycle (Bermúdez-Otero 2015) without encountering telescoping. Obviously, the way phonological processes age and whether or not they encounter telescoping has nothing to do with the cognitive system or the way it relates to the real world. Hence faithful mappings are not imposed by the cognitive system, which allows for any type of mapping, but are an artefact of external conditions: their existence and overwhelming frequency are due to the phonetic origin of phonological processes.

Given this backdrop, a similarity calculus of the kind described is outright impossible when two modules communicate within the cognitive system. It makes no sense to even think of similarity or non-similarity ("mismatch") when items such as gender, tense, number, person, animacy etc. are mapped onto units such as labial, occlusion, voice, etc. The relationship cannot be anything but arbitrary since the two vocabularies are incommensurable: any match is as unmotivated as any other.

A similarity calculus is possible, though, whenever a real-world signal is boxed into cognitive categories (a process called grammaticalization when language as a whole is concerned, phonologization when phonology is at stake). But this possibility is only due to the fact that analysts name the latter after the former. That is, the phonological category associated to the phonetic event of labiality is called [labial] because of its association to the real-world item – and for no other reason. But if this practice where phonetics defines both phonetic and phonological properties is done away with as in substance-free phonology where phonological objects identify as alphas, betas and gammas (see section 6.3), there is no similarity calculus possible even between phonetic labiality, occlusion etc. and phonological alphas, betas and gammas: a beta is not any more or less similar to labiality than it is to palatality.

In sum, thus, it is outright impossible to calculate similarity when two vocabularies are compared that both represent the cognitivized version of real-world items: tense, person, number, gender etc. are an exaptation of the real world just as much as labial, occlusion, voice, etc. When cognitive categories are matched with real-world items, though, the illusion of similarity may appear, but which is only due to the (unwarranted) practice of giving names to the former that come from the latter. That is, the names that analysts give to cognitive categories in order to be able to talk about them obfuscate their identity.

## CONCLUSION

It was argued on the preceding pages that computational translation does not qualify for intermodular communication because computational systems that can read two distinct vocabularies (Jackendoff's bi-domain specificity) are modular monsters. They defy the purpose of modularity (domain specificity) and make interface devices pointless: if modules can parse the vocabulary of their neighbors, no translation is needed in the first place.

If computational translation does not qualify, we fall back on the alternative that is comparatively well understood in the realm of language: lexical translation (list-based, look-up table) as practiced upon Vocabulary Insertion when morpho-syntactic structure is converted into phonological units. For this particular interface, the current standard of two channel translation where morphemic information is converted through a lexicon but boundary information by a computational means can be brought back to a uniform list-based interface device.

Beyond this particular interface and language more generally, it was shown that Gallistel & King's (2010) objection against lexical mechanisms, called the *infinitude of the possible*, is valid for modular computation where relevant functions may describe infinite sets of outputs (as is the case of the morpho-syntactic module), but does not concern intermodular communication where all sets of items encountered are finite by definition. This is because the to-be-translated items belong to domain-specific vocabulary that is stored in long-term memory: things that are stored cannot be infinite.

Lexical translation thus appears to qualify as a general mechanism that manages intermodular communication. It is also at play when cognitive categories are associated to real-world items (e.g. the list of matches between colors and wave lengths).

## REFERENCES

Athanasopoulos, P. (2009). 'Cognitive representation of colour in bilinguals: the case of Greek blues'. *Bilingualism: Language and Cognition*, 12, pp. 83-95.

Bach, E. & Harms, R. T. (1972). 'How do languages get crazy rules?', in Stockwell, R. & Macaulay, R. (eds.) *Linguistic change and generative theory.* Bloomington: Indiana University Press, pp. 1-21.

Bermúdez-Otero, R. (1999) *Constraint interaction in language change: quantity in English and German.* Ph.D dissertation, University of Manchester.

Bermúdez-Otero, R. (2003). 'The acquisition of phonological opacity', in Spenader, J., Eriksson, J. & Dahl, A. (eds.) *Variation within Optimality Theory: Proceedings of the Stockholm Workshop on Variation within Optimality Theory.* Stockholm: Department of Linguistics, Stockholm University [longer version at ROA #593], pp. 25-36.

Bermúdez-Otero, R. (2015). 'Amphichronic explanation and the life cycle of phonological processes', in Honeybone, P. & Salmons, J. C. (eds.) *The Oxford handbook of historical phonology.* Oxford: OUP, pp. 374-399.

Blaho, S. (2008) *The syntax of phonology. A radically substance-free approach.* Ph.D dissertation, University of Tromsø.

Blaho, S., Bye, P. & Krämer (eds.), M. (2007) *Freedom of analysis?* Berlin: Mouton de Gruyter.

Boersma, P. (1998) *Functional Phonology. Formalizing the interactions between articulatory and perceptual drives*. The Hague: Holland Academic Graphics.

Boersma, P. (2011) *A programme for bidirectional phonology and phonetics and their acquisition and evolution.* Ms, University of Amsterdam. ROA #868.

Boersma, P. & Hamann, S. (2008). ‘The evolution of auditory dispersion in bidirectional constraint grammars’. *Phonology*, 25, pp. 217-270.

Calvert, G. & Cambell, R. (2003). ‘Reading Speech from Still and Moving Faces: The Neural Substrates of Visible Speech’. *Journal of Cognitive Neuroscience*, 15, pp. 57-70.

Carruthers, P. (2006) *The Architecture of the Mind. Massive modularity and the flexibility of thought*. Oxford: Clarendon Press.

Carvalho, J. B. d. (2002). ‘Formally-grounded phonology: from constraint-based theories to theory-based constraints’. *Studia Linguistica*, 56, pp. 227-263.

Chabot, A. (2019). ‘What's wrong with being a rhotic?’. *Glossa*, 4, pp. article 38.

Chomsky, N. (2000). ‘Minimalist inquiries: the framework’, in Martin, R., Michaels, D. & Uriagereka, J. (eds.) *Step by Step. Essays on Minimalist Syntax in Honor of Howard Lasnik.* Cambridge, Mass.: MIT Press, pp. 89-155.

Chomsky, N. & Halle, M. (1968) *The Sound Pattern of English*. Cambridge, Mass.: MIT Press.

Choudhury, A. K. R. (2014) *Principles of colour appearance and measurement. Volume 1: Object appearance, colour perception and instrumental measurement*. Cambridge: The Textile Institute.

Cohn, A. (1998). ‘The Phonetics-Phonology Interface Revisited: Where's Phonetics?’. *Texas Linguistic Forum*, 41, pp. 25-40.

Collins, J. (2017). ‘Faculties and Modules: Chomsky on Cognitive Architecture’, in McGilvray, J. (ed.) *The Cambridge companion to Chomsky.* Cambridge: CUP, pp. 217-234.

Coltheart, M. (1999). ‘Modularity and cognition’. *Trends in Cognitive Sciences*, 3, pp. 115-120.

Dresher, E. (2014). ‘The arch not the stones: Universal feature theory without universal features’. *Nordlyd*, 41, pp. 165-181.

Dresher, E. (2018). ‘Contrastive Hierarchy Theory and the Nature of Features’. *Pro-*

*ceedings of the 35th West Coast Conference on Formal Linguistics*, 35, pp. 18-29.

Flemming, E. (2004). 'Contrast and perceptual distinctiveness', in Hayes, B., Steriade, D. & Kirchner, R. (eds.) *Phonetically-based Phonology.* Cambridge: CUP, pp. 232-276.

Fodor, J. (1983) *The modularity of the mind*. Cambridge, Mass.: MIT-Bradford.

Fodor, J. (2000) *The mind doesn't work that way: The scope and limits of computational psychology*. Cambridge, Mass.: MIT Press.

Gallistel, R. C. & King, A. P. (2010) *Memory and the computational brain. Why cognitive science will transform neuroscience*. Chichester: Wiley-Blackwell.

Gouskova, M. & Becker, M. (2016). 'Source-oriented generalizations as grammar inference in Russian vowel deletion'. *Linguistic Inquiry*, 47, pp. 391-425.

Grieve, K. W. (1991). 'Traditional beliefs and colour perception'. *Perceptual and Motor Skills*, 72, pp. 1319-1323.

Hale, K. (1973). 'Deep-surpface canonical disparities in relation to analysis and change: an Australian case', in Seboek, T. A. (ed.) *Diachronic, areal, and typological linguistics.* The Hague: Mouton, pp. 401-458.

Hale, M. & Kissock, M. (2007). 'The Phonetics-Phonology interface and the acquisition of perseverant underspecification', in Ramchand, G. & Reiss, C. (eds.) *The Oxford Handbook of Linguistic Interfaces.* Oxford: OUP, pp. 81-101.

Hale, M. & Reiss, C. (2000). 'Substance Abuse and Dysfunctionalism: Current Trends in Phonology'. *Linguistic Inquiry*, 31, pp. 157-169.

Hale, M. & Reiss, C. (2008) *The Phonological Enterprise*. Oxford: OUP.

Hamann, S. (2011). 'The Phonetics-Phonology Interface', in Kula, N., Botma, B. & Nasukawa, K. (eds.) *The Continuum Companion to Phonology.* London: Continuum, pp. 202-224.

Hamann, S. (2014). 'Phonetics-phonology mismatches', Paper presented at Old World Conference in Phonology, Leiden, 22-25 January.

Hansen, T., Olkkonen, M., Walter, S. & Gegenfurtner, K. R. (2006). 'Memory modulates color appearance'. *Nature Neuroscience*, 9, pp. 1367-1368.

Hayes, B., Kirchner, R. & Steriade (eds.), D. (2004) *Phonetically-Based Phonology*. Cambridge: Cambridge University Press.

Ingleby, M. & Azra, A. (2003). 'Phonological Primes and McGurk Fusion'. *Proceedings of the 15th International Congress of Phonetic Sciences*, pp. 2609-2612. WEB.

Inkelas, S. (1995). 'The consequences of optimization for underspecification'. *Proceedings of the North East Linguistic Society*, 25, pp. 287-302. [ROA #40].

Iosad, P. (2012) *Representation and variation in substance-free phonology. A case study in Celtic.* Ph.D dissertation, University of Tromsø.

Iosad, P. (2017) *A Substance-free Framework for Phonology. An Analysis of the Breton Dialect of Bothoa*. Edinburgh: Edinburgh University Press.

Itô, J. & Mester, A. (1999). 'Realignment', in Kager, R., Hulst, H. v. d. & Zonneveld, W. (eds.) *The prosody-morphology interface.* Cambridge: Cambridge University Press, pp. 188-217.

Jackendoff, R. (1987) *Consciousness and the computational mind*. Cambridge, Mass.: MIT Press.

Jackendoff, R. (1992) *Languages of the mind*. Cambridge, Mass.: MIT Press.

Jackendoff, R. (1994) *Patterns in the Mind. Language and human nature*. New York: BasicBooks.

Jackendoff, R. (1997) *The Architecture of the Language Faculty*. Cambridge, Massachusetts: MIT Press.

Jackendoff, R. (2002) *Foundations of Language. Brain, Meaning, Grammar, Evolution*. Oxford: Oxford University Press.

Keating, P. (1985). 'Universal phonetics and the organization of grammars', in Fromkin, V. (ed.) *Phonetic Linguistics Essays in Honour of Peter Ladefoged.* Orlando: Academic Press, pp. 115-132.

Kingston, J. (2019). 'The interface between phonetics and phonology', in Katz, W. F. & Assmann, P. F. (eds.) *The Routledge Handbook of Phonetics.* Abingdon: Routledge, pp. 359-400.

Marr, D. (1982) *Vision*. San Francisco: Freeman.

McCarthy, J. (1998). 'Morpheme structure constraints andparadigm occultation', in Gruber, M. C., Higgins, D., Olson, K. & Wysocki, T. (eds.) *Chicago Linguistics Society 32. Part 2: The Panels.* Chicago: Chicago Linguistic Society, pp. 123-150.

McCarthy, J. & Prince, A. (1993). 'Generalized Alignment', in Booij, G. & Marle, J. v. (eds.) *Yearbook of Morphology 1993.* Dordrecht: Kluwer. Abridged version in McCarthy, John (ed.) 2004. Optimality Theory in Phonology, 451-463. Oxford: Blackwell, pp. 79-153.

McCarthy, J. & Prince, A. (2001) *Prosodic Morphology. Constraint Interaction and Satisfaction.* Ms, ROA #482.

McGurk, H. & MacDonald, J. (1976). 'Hearing Lips and Seeing Voices'. *Nature*, 264, pp. 746-748.

Mielke, J. (2008) *The Emergence of Distinctive Features*. Oxford: OUP.

Nespor, M. & Vogel, I. (1986) *Prosodic Phonology*. Dordrecht: Foris.

Odden, D. (2006). 'Phonology ex nihilo, aka Radical Substance-Free Phonology and why I might recant', Paper presented at Phonological seminar, Tromsø, 6 December.

Odden, D. (forth). 'Radical Substance Free Phonology and Feature Learning'. *To appear in the Canadian Journal of Linguistics*.

Peperkamp, S. (1995). 'Prosodic constraints in the derivational morphology of Italian'. *Yearbook of Morphology 1994*, pp. 207-244.

Prince, A. & Smolensky, P. (1993 [2004]) *Optimality Theory. Constraint Interaction in Generative Grammar.* Ms, Rutgers University, University of Colorado (ROA version August 2002). Revised version published by Blackwell in 2004.

Prince, A. & Smolensky, P. (2004 [1993]) *Optimality Theory. Constraint Interaction in Generative Grammar*. Oxford: Blackwell.

Rasin, E. (2018) *Modular Interactions in Phonology.* PhD dissertation, MIT.

Rasin, E. (Ms. (2016)) *Morpheme structure constraints and blocking in nonderived en-*

*vironments*. Ms., MIT.

Rasin, E. & Katzir, R. (2015). 'A learnability argument for constraints on underlying representations', in Bui, T. & Özyildiz, D. (eds.) *Proceedings of the 45th Annual Meeting of the NorthEast Linguistic Society (NELS), Vol.2.* Cambridge, Mass.: GLSA, pp. 267-288.

Samuels, B. (2012). 'The emergence of phonological forms', in Di Sciullo, A. M. (ed.) *Towards a biolinguistic understanding of grammar. Essays on interfaces*. Amsterdam: Benjamins, pp. 193-213.

Scheer, T. (2008). 'Why the Prosodic Hierarchy is a diacritic and why the Interface must be Direct', in Hartmann, J., Hegedüs, V. & van Riemsdijk, H. (eds.) *Sounds of Silence: Empty Elements in Syntax and Phonology.* Amsterdam: Elsevier, pp. 145-192. WEB.

Scheer, T. (2011) *A Guide to Morphosyntax-Phonology Interface Theories. How Extra-Phonological Information is Treated in Phonology since Trubetzkoy's Grenzsignale*. Berlin: Mouton de Gruyter.

Scheer, T. (2012) *Direct Interface and One-Channel Translation. A Non-Diacritic Theory of the Morphosyntax-Phonology Interface. Vol.2 of A Lateral Theory of phonology*. Berlin: de Gruyter.

Scheer, T. (2014). 'Spell-Out, Post-Phonological', in Cyran, E. & Szpyra-Kozlowska, J. (eds.) *Crossing Phonetics-Phonology Lines.* Newcastle upon Tyne: Cambridge Scholars, pp. 255-275.

Scheer, T. (2019). 'Phonetic arbitrariness: a cartography'. *Phonological Studies*, 22, pp. 105-118.

Scheer, T. (forth). '3xPhonology'. *To appear in the Canadian Journal of Linguistics*.

Selkirk, E. (1981 [1978]). 'On prosodic structure and its relation to syntactic structure', in Fretheim, T. (ed.) *Nordic Prosody II.* Trondheim: TAPIR, pp. 111-140.

Selkirk, E. (1984) *Phonology and Syntax: The Relation between Sound and Structure*. Cambridge, Mass.: MIT Press.

Selkirk, E. (1986). 'On derived domains in sentence phonology'. *Phonology*, 3, pp. 371-405.

Selkirk, E. (1996). 'The prosodic structure of function words', in Morgan, J. & Demuth, K. (eds.) *Signal to syntax: bootstrapping from syntax to grammar in early acquisition.* Mahwah, NJ: Erlbaum, pp. 187-213.

Sperber, D. (2001). 'In Defense of Massive Modularity', in Dupoux, E. (ed.) *Language, Brain and Cognitive Development.* Cambridge, Mass.: MIT Press, pp. 47-57.

Steriade, D. (1999). 'Phonetics in Phonology: The case of Laryngeal Neutralization'. *UCLA Working Papers in Linguistics*, 2, pp. 25-146.

Stevens, K. A. (2012). 'The vision of David Marr'. *Perception*, 41, pp. 1061-1072.

Vaux, B. (2005). 'Formal and empirical arguments for Morpheme Structure Constraints', Paper presented at Linguistic Society of America, Oakland, January 5th (written version Ms 2011).

Volenec, V. & Reiss, C. (2018). 'Cognitive Phonetics: The Transduction of Distinctive Features at the Phonology–Phonetics Interface'. *Biolinguistics*, 11, pp. 251-294.

Volenec, V. & Reiss, C. (2019). 'Formal Generative Phonology'. *Radical*, 1, available at http://radical.cnrs.fr/volenec-reiss-formal-generative-phonology/.

Webster, M. A. (2003). 'Light adaptation, contrast adaptation, and human colour vision', in Mausfeld, R. & Heyer, D. (eds.) *Colour Perception. Mind and the physical world.* Oxford: OUP, pp. 67-110.

Whittle, P. (2003). 'Contrast Colours', in Mausfeld, R. & Heyer, D. (eds.) *Colour Perception. Mind and the physical world.* Oxford: OUP, pp. 115-139.

Winawer, J., Witthoft, N., Frank, M. C., Wu, L., Wade, A. R. & Boroditsky, L. (2007). 'Russian blues reveal effects of language on color discrimination'. *PNAS*, 104, pp. 7780-7785.

Yip, M. (1996). 'Lexicon Optimization in languages without alternations', in Durand, J. & Laks, B. (eds.) *Current trends in Phonology. Models and Methods, Vol.2.* Salford, Manchester: ESRI, pp. 759-790.

Yip, M. (1998). 'Identity Avoidance in Phonology and Morphology', in Lapointe, S., Brentari, D. & Farrell, P. (eds.) *Morphology and its Relation to Phonology and Syntax.* Stanford: CSLI Publications, pp. 216-246.

## DISCUSSION WITH GABRIEL BERGOUNIOUX

**(UNIVERSITÉ D'ORLÉANS, CNRS/LLL)**

Bergounioux, Gabriel. 2020. discussion in: Scheer, Tobias (auth.) "On the lexical character of intermodular communication". *Radical: A Journal of Phonology*, 1, 223-228.

### COMMENTS

**Objet de la contribution.** Dans une perspective historique couvrant un large spectre, l'auteur propose de considérer l'interface de la phonologie et de la morphologie, conçues en termes de *modules*, comme relevant possiblement de deux mécanismes :

- lexical, à partir de la morphologie (insertion de vocabulaire),
- computationnel, à partir de la phonologie.

Pour Scheer, le mécanisme computationnel n'est pas à même de supporter le passage d'un module à l'autre. En prenant pour argument l'irrecevabilité des composants lexicaux dans les catégories de la phonologie, il en déduit une dissymétrie : alors que le lexique est à même d'intégrer les données phonologiques, l'inverse n'est pas soutenable. Dès la présentation, deux questions sont corrélées en prolongement du thème traité qui feront l'objet de développements particuliers :

- l'une implique la faculté de langage,
- l'autre la correspondance cognitive de ce postulat.

La discussion de cet article est restreinte à la seule hypothèse de départ, laquelle conditionne les conclusions à tirer pour les autres points.

**Le choix du modèle.** Alors qu'en introduction il était fait mention de toute discussion sur la relation entre phonologie et morphologie, les premières lignes de l'article en réduisent l'examen aux études conduites depuis *SPE*, entérinant le postulat chomskyen de « modules » qui implique un traitement des relations en termes d'interface. C'est l'une des trois solutions majeures proposées dans l'histoire de la phonologie, chacune pouvant être sténographiée d'un nom, d'une œuvre et d'une figuration :

- Saussure, *Mémoire* (1879) : une topologie en bande de Moebius où la conversion de données phonologiques en morphologie (apophonie) et réciproquement (syllabation) opère de façon continûment réversible ;
- Troubetzkoy, *Grundzüge* (1939) : une hiérarchie ascendante, par strates, de la phonétique à la phonologie, de la phonologie à la morphologie etc. chacune intégrant les données de la précédente et les traitant dans ses propres catégories, une perspective reprise par Jakobson ;
- Chomsky, *SPE* (1968) : trois modules autonomes en interaction (interfaces).
- OT et LabPhon se sont plutôt inspirés de la deuxième représentation tandis que la troisième est illustrée, entre autres, par Scheer qui en exploite le modèle dans tout son exposé.

Dans ce modèle, « the lexicon is a list of items stored in long term memory which matches phonological units with morpho-syntactic properties », ce qui revient à soutenir (i) que les entités sont conçues directement comme des entrées lexicales et non au niveau de la concaténation des morphèmes, (ii) que sont dissociés par principe la phonologie et la morphologie (une conséquence de la modularité). Cette conception est partagée par toutes les théories de la « Two-Channel Translation ».

En considérant la variante américaine du structuralisme, Scheer conclut à une indifférence de cette théorie pour les effets de la morphologie sur la phonologie auquel ont remédié la notation # et le « Readjustment Component » de *SPE*. Les contributions de la

Prosodic Phonology – dans le paradigme autosegmental – et d'OT – en théorie des contraintes – sont exposées en une trentaine de lignes (avec référence à d'autres travaux de l'auteur pour une discussion détaillée) avant que ne soit examinée la modularité sous le titre peu engageant de « Modular Monsters ».

**De Fodor à Jackendoff, ce que sont censés faire les modules.** Fodor pose en principe que dans « [the] domain specificity and the modular approach (…) computational systems can only understand and parse one single type of vocabulary, the one over which it operates »[12] (p. 191), ce qui pose la question du traitement des emprunts dans le module phonologique.

Pour Jackendoff, la « Structure Constrained Modularity » se fonde sur un modèle de l'activité mentale distinguant trois types de modules (ou « processeurs ») :

- d'inférence, qui est d'ordre logique et cognitif,
- d'intégration, qui est d'ordre sensoriel et symbolique,
- d'interface.

Appliqué au langage, un schéma en est proposé p. 195. Il postule des correspondances entre modules sans autre exigence que celle d'une apprenabilité, en soulignant qu'une partie de chaque module est accessible à une partie de celui avec lequel il entre en relation. Les processeurs d'interface résultent d'un assemblage d'éléments empruntés aux deux domaines. Selon Scheer, cette conception est contradictoire avec le principe de modularité en ce que, *in fine*, elle ne nécessiterait plus d'interface.

**Le choix de l'accès lexical.** Après avoir écarté les théories non modulaires puis refusé des conceptions modulaires qui, soit ne pourraient avoir accès à une partie des données (Fodor et l'emprunt), soit attribueraient aux modules des propriétés telles qu'elles ren-

12 [editor's note: the quoted sentence has been modified in the last version of the paper]

draient superflue l'interface (Jackendoff), Scheer fait l'hypothèse que « If computational translation does not qualify in a modular environment, the only alternative available to date appears to be a lexical access » (p. 197).

La restriction permet de clarifier les exigences de la démonstration. Chaque module est accessible à plusieurs autres qui présentent eux-mêmes cette propriété d'interaction. L'exemple est donné de l'audition traitée comme un module en prenant pour exemple l'effet McGurk. La démonstration se focalise sur l'acquisition de l'information sensorielle plutôt que sur l'implémentation des structures phonologiques ou sur les projections que fait l'auditeur en fonction de ses attentes pragmatiques dans l'échange. Dans l'effet McGurk, les résultats sont d'autant plus probants qu'ils sont recueillis en situation expérimentales et qu'ils recourent à l'émission de logatomes.

Avec un usage du terme « lexique » qui unifie deux acceptions, celle générique d'un vocabulaire à disposition du locuteur, et celle spécifique qui réfère à l'ensemble des unités de concaténation actives dans un module (le recouvrement est massif mais non identique), un schéma présente, de nouveau à partir de la réduction phonologique d'un signal de parole, une modélisation de la communication intermodulaire à accès lexical. Un exemple concret aurait permis d'apprécier, pour chacune des étapes figurées sur le graphe de la page 199, le réalisme ou non du modèle. S'il ne paraît pas nécessaire de procéder à un traitement à grande échelle des données pour étayer la démonstration, le caractère abstrait du raisonnement ne facilite pas la vérification par l'épreuve d'un raisonnement sur pièce.

**L'arbitraire.** L'arbitraire est l'un des arguments fondateurs de la modularité, i.e. de la nécessité de concevoir une interface puisque rien de « naturel » ne relie entre eux les composants du langage. Les marques métalinguistiques nécessitées par le traitement, en particulier l'utilisation de diacritiques, sont exclues du lexique conçu à la fois comme un répertoire – mémorisé par le locuteur – et comme la représentation formelle qu'en construit le linguiste. Les notations symboliques sont éliminées du lexique au profit d'une « interface directe ». Scheer oppose les symboles de segmentation et de clustering de la morphologie « au vocabulaire phonologique ». Celui-ci est identifié moins comme le produit d'un système d'oppositions qui décident de la détermination des unités que, dans un retour au substrat sonore, comme la combinaison de traits de substance phonétique.

L'argument de l'arbitraire est repris au moment de procéder à un examen de la relation entre la phonétique et la phonologie. Les distorsions entre les représentations sous-jacentes et les réalisations concrètes, quantifiées à 1%, servent de justification à la production d'un nouveau schéma (p. 203). La proposition d'une phonologie « substance free », où les unités sont notées par des symboles littéraux, préserve deux propriétés liées à la substance : la primauté des unités segmentales sur leurs relations, une différenciation obtenue par contraste sur lesplutôt que par opposition.

**Conclusion de l'article.** La conclusion de Scheer réaffirme l'hypothèse de départ : « lexical translation (list-based) as practiced upon Vocabulary Insertion when morpho-syntactic structure is converted into phonological units », i.e. la réaffirmation d'une indépendance, dans les langues, de leur morphologie et de leur phonologie, i.e. ce qui fonde dans son principe la modularité. Pour compenser les effets de coupure entre modules, il est nécessaire d'introduire des interfaces qui sont étendues, comme cet article en donne l'exemple, à la relation entre le signal langagier et la phonologie d'un côté, la morpho-syntaxe et les compétences cognitives de l'autre.

**Eléments de discussion.** Cette contribution porte sur l'architecture même du langage et reprend à son compte une hypothèse cardinale (et ses conséquences) que partagent les théories issues du générativisme : toute langue est constituée de trois modules qui assurent globalement l'interface entre la parole et les représentations mentales. L'autonomie de chacun de ces modules impose que soit apportée une explication quant à leur mode de relation.

La modularité n'est pas, dans le paradigme structural, une théorie construite par les linguistes à partir de la description des langues. Elle a été importée du champ des sciences de l'esprit où elle renouvelait les discussions initiées sur les localisations cérébrales en se fondant sur des techniques instrumentales d'investigation. L'adaptation à cette construction symbolique qu'est le langage impose de prendre en compte, dans la discussion, l'ensemble des modèles en concurrence. Pour cela, un certain nombre d'exemples cruciaux, des *benchmarks*, permettraient d'évaluer, à supposer que la comparabilité des modèles soit possible, les qualités de chacun. La démonstration de Scheer est explicite quant aux insuffisances qu'il décèle dans un modèle modulaire mais sans prendre en compte que la modularité n'est pas la seule représentation possible et sans rien dire d'éventuelles difficultés que rencontrerait le schéma qu'il propose.

L'auteur s'en explique en justifiant la concision de certaines démonstrations par des références à certaines de ses publications. Il serait plus convaincant d'exposer des exemples cruciaux qui fourniraient matière à une discussion où seraient à même d'être mesurées les différentes hypothèses. La falsifiabilité constitue l'une des exigences majeures du débat scientifique et si aucun des arguments ne serait réfutable aussi longtemps qu'on admettrait les prémisses de Scheer, ceux-ci pourraient être remis en case par le choix d'un autre cadre interprétatif.

## REPLY

See reply p. 232.

## DISCUSSION WITH FLORIAN BREIT

**(UNIVERSITY COLLEGE LONDON)**

Breit, Florian. 2020. discussion in: Scheer, Tobias (auth.) "On the lexical character of intermodular communication". *Radical: A Journal of Phonology*, 1, 229-239.

### COMMENTS

**Summary.** The paper brings together various arguments from a long line of the author's own previous work on the linguistic interfaces and modularity, especially Scheer (2011, 2012, 2014). The main argument is this: if we accept that domain-specificity means a module can only process a single set of proprietary symbols, and that domain-specificity is a property of all modules (specifically excluding central processes from this term), then it follows that the only plausible interface mechanism is an arbitrary list-based translation mechanism. The author also argues that this applies both to intermodular interfaces and to transduction at the external perceptual interfaces (e.g. information originating from the neural signal produced by macular cone cell stimulation).

The author goes on to discuss how the modular, list-based translation mechanism relates to a number of issues such as the involvement of computation at the interface in SPE (Halle & Chomsky 1968) or the interface processors proposed by Jackendoff (1997). While, for the most part, Scheer has already made these arguments extensively in his previous work, put together here they form a suitable and persuasive backdrop for Scheer's defence of the list-based translation view against a series of posts published last year by Bill Idsardi on the Faculty of Language blog. In essence, Scheer argues that Idsardi's first criticism of intractability is not applicable to the interface while his second criticism, namely that list-based arbitrary translation would seem to be unique to language (rather than cognition at large), by giving a number of examples of other mapping mechanisms which could be captured in a similar vein, principally colour categorisation.

**Domain-specificity diagnoses modularity.** While I find myself in content agreement with the description of the role of domain-specificity and its consequences in Section 3, I have the impression (perhaps mistakenly) that a possible reading of the way the issues is presented here would be that domain-specificity essentially subsumes informational encapsulation. That is, when Scheer says "[a module] is incompetent for, or insensitive to anything that is done in neighbouring modules: it cannot parse or understand information that is foreign of its own domain", I see one interpretation where modules cannot refer to anything that happens in another module, but they might yet make reference to something that happens in central systems, and another possible interpretation, putting more weight on the latter clause, where domain-specificity implies that a module cannot refer to any module-external (and hence not part of the module's proprietary vocabulary) information whatever, to include central systems.

This seems to me an interesting issue, given especially that e.g. Pylyshyn (1999), who in turn appears to at least partially subsume restricted access under informational encapsulation, sees informational capsulation as also being a typically characteristic of modules as opposed to central systems, similar to the special role taken here by domain-specificity. Breit (2019) similarly argues that a combination of the properties of domain-specificity, informational encapsulation, and restricted access is responsible for the maximal modularity effects we see in the case of modules such as morpho-syntax and phonology.

Why is the differential interpretability of Scheer's description so interesting? Because if it implies that domain-specificity only excludes the possibility of a module being influenced by processes/knowledge in another *module*, but not by non-modular processes or information that form part of the central systems, this might serve as an outward discrimination criterion for modularity. For instance, Tanenhaus et al. (1995) show that visual discourse context is used to resolve structural ambiguity in sentence parsing, evidence which is classically interpreted as showing that the syntactic parsing process is not informationally encapsulated (since its processing seems to make reference to information otherwise not represented by its own proprietary vocabulary). If domain-specificity now independently rules out that the syntactic parser can refer to information in other (hypothetically unrestricted) modules, then we can draw a conclusion from evidence on what the parser does about the modularity of the component responsible for the interpretation of visual discourse (presumably pragmatics), which consequently must be part of the central system and cannot be a module. Conversely, if the latter view is correct, and domain-specificity also rules out that a module can make reference to processing in the central systems, then this would imply that the parser itself cannot be modular, but doesn't tell us anything about the (non-)modularity of pragmatics.

**Against multi-domain specificity means for phonetic arbitrariness.** Scheer's argument against Jackendoff's (1997) proposal of bi-domain specificity for interface processors in Section 4.3 is interesting also in that it seems to constitute an argument not only against bi-domain specificity, but also against the idea, endemic to much SPE-related work, that the task of phonology is to somehow take as it's input a "phonological form" and then render as its outcome a "phonetic form". Because there is no multi-domain specificity, modules can only map like onto like, and it follows that the task of phonology is simply to map phonological objects onto phonological objects. While of course this argument is not novel, and a fundamental assumption of work in the Government Phonology tradition (cf. e.g. Kaye 1995), it is notable that the argument itself can seemingly be derived quite directly from the notion of domain-specificity. We see this surface of course later in Section 6.3, where Scheer lays out under the model he proposes, stepping from phonological forms to phonetic processing must involve arbitrary list-based translation, similar to translation at any other modular interface, a view that we see very slowly entertained in more corners of the field.

**Breit**, F. (2019) *Welsh Mutation and Strict Modularity*. PhD thesis, University College London. **Halle**, M. & **Chomsky**, N. (1968) *The Sound Pattern of English*. New York, NY: Harper & Row. **Jackendoff**, R. S. (1997) The Architecture of the Language Faculty. *Linguistic Inquiry Monographs*. Cambridge, MA: MIT Press. **Kaye**, J. (1995) Derivations and Interfaces. *Frontiers of Phonology*, edited by Jacques Durand & Francis Katamba, 289-332. London & New York: Longman. **Pylyshyn**, Z. W. (1999) *Computation and cognition: toward a foundation for cognitive science*. Cambridge, MA: MIT Press. **Scheer**, T. (2011) *A Guide to Morphosyntax-Phonology Interface Theories. How Extra-Phonological Information is Treated in Phonology since Trubetzkoy's Grenzsignale*. Berlin: Mouton de Gruyter. **Scheer**, T. (2012) *Direct Interface and One-Channel Translation. A Non-Diacritic Theory of the Morphosyntax-Phonology Interface. Vol. 2 of A Lateral Theory of Phonology*. Berlin: Mouton de Gruyter. **Scheer**, T. (2014). 'Spell-Out, Post-Phonological', in Cyran, E. & Szpyra-Kozlowska, J. (eds.) *Crossing Phonetics-Phonology Lines*. Newcastle: Cambridge Scholars, pp. 255-275. **Tanenhaus**, M. K., Spivey-Knowlton, M. J., Eberhard, K. M. & Sedivy, J. C. (1995) Integration of Visual and Linguistic Information in Spoken Language Comprehension. *Science*, 268.

## REPLY

The process from manuscript to published article that Radical puts to use has, among other things, the interesting feature that the reviewing process, which is ordinarily anonymous and invisible to the audience, is disclosed to a certain extent: both reviewers and the author can talk about the arguments exchanged, or continue the discussion, in an appendix to the article. Radical's idea is that this exchange is an integral part of the article and the work that people afforded to get it published.

I would like to take advantage of this double opportunity here: both to disclose some of the exchange during the reviewing process with Breit, and to continue the discussion with him.

Here are five statements which I believe are correct.

**1. What domain specificity is.** Domain specificity means that computational systems in the mind/brain that have modular status can only parse a specific vocabulary, to the exclusion of all others. Their input is restricted and hence they are unable to parse any other items, of whatever origin (other modules, central systems, my pinkie or the planet mars).

**2. Domain specificity is consubstantial with modules.** There are no modules, and hence there is no modularity, without domain specificity. And there is no domain-specificity without modules. The very idea that Fodor expressed in his 1983 book (which condensed 19th century phrenology and work such as Lashley's 1951), its raison d'être, is that computation carried out in the mind is not all-purpose but falls into a number of specialized systems that are designed to carry out a specific task. The subtitle of Fodor's book is "an essay on faculty psychology": faculties are distinct and specific. The competing approach to how the mind works is connectionism. Since its founding statement in 1986, connectionism holds that there is only one type of computation in the mind that underlies all functions: computation is colourless and hence domain-general. This is the exact negation of Fodor's conception of the mind: it holds that the input to computation is not restricted, while Fodor says it is. Asserting that there is a version of modularity where computational systems can parse information that belongs to more than one domain is a contradiction in terms.

**3. Fodor has really said all that, and he has really meant it.** The preceding is not just a specific interpretation of Fodorian modularity. Yes, there are a number of properties that have been argued to characterize modules, and there are different views on that. No, the massive literature on modularity does not contain any view whereby modules process more than one vocabulary: that would be a contradiction in terms, see point two. Only linguists appear to produce this type of contradiction in terms: Jackendoff's bi-domain specificity and Idsardi's cross-module veridicality calculus are cases in point (which are discussed below).

**4. Fodor didn't leave any door open.** Fodor has never left a door open for modules to be able to parse foreign vocabulary. The footnote on the McGurk effect in Fodor (1983), hinted at by Idsardi and discussed in section 7.1 of the article, is not evidence in favour of a door having been left open, or of Fodor being unsure. On the contrary it shows that Fodor has double-locked this door.

**5. Modules can take input from anybody, provided it is translated.** Given domain specificity, modules can only take in information from other entities if this information appears in their own proprietary vocabulary. The question is not whether modules can communicate with other modules, central systems, my pinkie or the planet mars – of course they can, provided whatever information is sent reaches the receiving module in form of its proprietary vocabulary. If my pinkie works with phonological vocabulary, it can directly send information to the phonological module, which will parse this information without translation and start its computation based on this input. If my pinkie has no sense of the vocabulary used in the phonological module, it may send information but this information will only be parsed if it is first translated into the phonological vocabulary. Domain-specificity does not entail that modules are autistic or cannot communicate with other entities – it entails that there is no communication with other modules without prior translation. If my pinkie or a central system are able to send information coded in phonological vocabulary, no translation is needed. Other modules are not able to air information in the phonological vocabulary since they work on a distinct vocabulary.[13]

It was unexpected and surprising to me to see that after almost 30 years of discussion since the publication of Fodor's book, I had to go through an argument about domain specificity. Not about whether or not it is correct, but about what it means. Breit said

13 Whether central systems are able to issue information in proprietary vocabularies of individual modules, and if so, whether they actually do that, is a separate question that has no bearing on the present discussion. It touches on Massive Modularity (Sperber 2001), i.e. the idea that central systems ultimately also identify as modules, which according to Fodor 1987: 27 is the "modularity thesis gone mad".

that maybe it doesn't mean what it means (statement 1), that maybe modularity is conceivable and is actually conceived without domain specificity (statement 2), that maybe some modules may be domain-specific but others may not (statement 2), and that Fodor hasn't really said that domain-specificity is a necessary property of modules (statements 3 and 4). I thought that in 2019, the field being minimally cumulative, it was enough to introduce domain specificity with a reference or two in order to be able to talk about its consequences (just as I would introduce rule ordering with a reference to SPE). Breit told me that I may have a private understanding of domain specificity and that the audience should be made aware of that: "it is not made clear in the text that this is the author's own interpretation and assertion, rather than the generally accepted view on Fodorian modularity." This of course curtails the argument made in the article, since somebody can adhere to modularity, but not to the private views of Scheer – which in fact are not endorsed by the field. Since the argument in the article is based on Scheer's private views, it can be dismissed by dismissing the premise.

I thought I had read enough Fodor and about Fodor, as well as on modularity. So I went back to the relevant literature, trying to find out about this "generally accepted view on Fodorian modularity" where modules may not abide by domain specificity and which I have missed out on. Nothing of the kind was in sight, though: handbook after handbook, article after article, all confirmed what I thought modularity was. Did Breit quote references where the "generally accepted view on Fodorian modularity" is exposed? Yes, exactly one: Coltheart (1999). This article discusses a number of characteristics that modules have been argued to possess, and the author concludes that none of them is obligatorily present in an item in order for that item to be called a module. Except one: domain specificity. That is, nothing can be a module that is not domain-specific. Coltheart (1999: 115) writes: "I argue that it is nevertheless possible to derive a useful definition of modularity from the kinds of arguments put forward by Fodor: A cognitive system is modular when and only when it is domain-specific." This is thus the one single reference that Breit provided in support of the idea that domain specificity is not a necessary property of modules. I asked him for other references supporting his view, or laying out the workings of a modular system where modules are not domain specific – to no avail. Section 3.1 now reflects this exchange with Breit: instead of introducing domain specificity with a reference or two, it is explained from scratch with quotes from relevant literature.

Our exchange is also reflected in Breit's published comments on the article, whose first point is this: "if we accept that domain-specificity means a module can only process a single set of proprietary symbols, and that domain-specificity is a property of all modules (specifically excluding central processes from this term), then it follows..." The introductory "if" shows that Breit still entertains the option that there are, or could be, modules which parse more than one type of vocabulary (statement one), and modules which are not domain-specific (statement two). Hence the reader will understand that these alternative views are entertained somewhere, but like in my exchange with Breit, is not given any hint at who or where somebody has said that.

Another issue discussed by Breit in his published comments is the evidence showing that visual discourse context may resolve structural ambiguity in sentence parsing. In the view of some authors, this shows that the syntactic module is not encapsulated, since it accepts foreign influence. This appears to be a misunderstanding. Encapsulation is not about whether or not a module can communicate with other modules (or central systems). Of course it can and does (statement five). Encapsulation says that once the module has begun a computation based on a given input, there is no way either information can leave the module (intermediate steps), or new information can be taken into account, before the running computation is completed. Nothing can get in or out during a computation. In syntax this is called inclusiveness: a numeration works on those pieces that were present when it started, and won't take any others into account (Chomsky 1995: 228). Hence a perfectly encapsulated module such as syntax is of course able to be influenced by vision, provided the visual vocabulary is translated prior to reaching the door of syntax. This is the exact same misunderstanding as the one discussed in section 7.1 of the article regarding Fodor's (1983) footnote on the McGurk effect: the fact that vision can bear on language does not mean that both domains process the same vocabulary and hence that there is no domain specificity (or encapsulation). It just means that intermodular communication affords translation.

In his published comments, Breit also distinguishes between the ability of modules to receive untranslated information from other modules vs. from non-modules (central systems). In Breit's interpretation, the former is ruled out by domain specificity (modules cannot talk to other modules without translation), while the latter may not (central systems may talk to modules without translation). This distinction is irrelevant, though (statement 5): domain specificity rules out the fact that modules parse vocabulary that is not theirs, no matter where it comes from.

Interestingly, Breit is not the only linguist engaged in modularity who tampers with domain specificity. Jackendoff's case is discussed in section 4 of the article: he not only stands on modular grounds, but also (unlike Breit) does not doubt that domain specificity is a necessary properties of modules. But what he calls bi-domain specificity, a property of interfaces in his view, is a contradiction in terms since it allows (interface) modules to parse more than one type of vocabulary. Bi-domain specificity holds up the word *domain specificity* but makes it an empty shell that now means the very reverse. Finally, Idsardi's idea to extend the competence of modules to more than just one domain (vocabularies) is discussed in section 7.1 of the article: while Jackendoff restricts such multi-vocabulary systems to interface modules, Idsardi suggests that regular modules could also be of this kind.

Why is it that linguists, who work in the modular framework, endorse its premises and aim at furthering its development, dissolve its essence? The anti-modular camp will enthusiastically applaud: connectionists have always said that computation in the mind is domain-general and that there are no specialized units whose input is restricted to a specific vocabulary. They receive unexpected support from within the modular camp and will pleasantly note that even Fodorian followers have come to reason, showing that the whole idea of specialized computational systems dissolves when put to use.

I do not contend that this is Breit's, Jackendoff's or Idsardi's goal, but it may be the effect of what they do. So why are they prepared to take that risk if they are otherwise engaged in the modular enterprise? For Breit I don't know: in his PhD (Breit 2019) he argues against accounts of Welsh mutation that rely on the presence of untranslated morpho.syntactic information in phonology and proposes an analysis abiding by domain specificity where phonology only parses translated morpho-syntactic information. For Jackendoff and Idsardi the answer is obviously their interface theory. As discussed in the article, since his earliest writings on the matter, Jackendoff believes that translation is computational in kind. If this is the case, then computational devices that carry out translation are necessarily able to parse the vocabularies of the two modules that they connect. Hence Jackendoff has no choice: interfaces must parse several vocabularies because they are computational. This is the reason, I argue in the article, why computational translation is incorrect.

Idsardi's interface management on the other hand is based on a similarity calculus between items that belong to distinct modules (section 7.5 of the article). Mappings from phonology to phonetics are faithful most of the time (phonological labiality is externalized as phonetic labiality), but sometimes are not (the phonological sonorant /r/ is pronounced as a uvular obstruent in French for example). This is what he calls partial veridicality. Like Jackendoff's system, Idsardi's supposes a mechanism that relates distinct modules and can see into both, hence can parse both vocabularies: to translate with Jackendoff, to carry out the similarity calculus with Idsardi.

The fact that these interface theories stand the very purpose of modularity on its head may be good reason, for those who believe that computation in the mind/brain is not all-purpose but done by specialized systems, to think of alternatives. Lexical translation is a well-known acquaintance (within language and beyond) that does not make connectionists applaud.

**Breit**, F. (2019) *Welsh Mutation and Strict Modularity*. PhD thesis, University College London. **Chomsky**, N. (1995) *The Minimalist Program. Cambridge*, MA: MIT Press. **Coltheart**, M. (1999). 'Modularity and cognition'. *Trends in Cognitive Sciences*, 3, pp. 115-120. **Fodor**, J. (1987). 'Modules, Frames, Fridgeons, Sleeping Dogs, and the Music of the Spheres', in Garfield, J. (ed.) *Modularity in knowledge representation and natural language understanding*. Cambridge, Mass.: MIT Press, pp. 25-36. **Lashley**, K. S. (1951). 'The problem of serial order in behavior', in Jeffress, A. L. (ed.) *Cerebral Mechanisms in Behavior: The Hixon Symposium*, New York: Wiley, pp. 112-136. **Sperber**, D. (2001). 'In Defense of Massive Modularity', in Dupoux, E. (ed.) *Language, Brain and Cognitive Development*. Cambridge, Mass.: MIT Press, pp. 47-57.