Tobias Scheer

Abralin ao vivo
14 November 2020
2:00 pm Paris time

Keywords:
modularity, interfaces, diacritics, substance-free phonology, phonetic arbitrariness

Long version

Abstract
Modules, interfaces and their consequences

The talk shows what a modular architecture of language could look like, what predictions it makes and what consequences it has. The idea that the mind (and the brain) are made of a number of distinct and functionally specialized systems (domain-specificity) is the founding statement of modularity, introduced by Fodor (1983) in modern times (building on 19th century phrenology).

If there are a number of distinct systems (modules) that work on specialized vocabulary that is unintelligible to neighbours, it follows that inter-modular communication can only occur through a translation from one into another vocabulary. This is what linguists call interfaces. On this backdrop, the talk addresses the following issues:

1. Pieces concernced
   The central piece of the language-related system (in production): long term memory → concatenation (morpho-syntax) → phonology → phonetics → extra-linguistic items (motor control etc.). Other systems such as semantics or pragmatics are not discussed.

2. Interfaces: lexical translation
   What does an interface look like on modular standards? It is argued that translation among two modules, called spell-out when morpho-syntactic structure is converted into phonological material, cannot be computational in kind (as in Ray Jackendoff's work) but must rather be list-based (lexical): it involves a dictionary-based lexical access where items are retrieved from long term memory.

3. Modules use only one mechanism to talk to each other
   This interface mechanism is the same all through, i.e. relating morpho-syntax and phonology as much as phonology and phonetics. That is, if you know how the former interface works, you know what the latter looks like.

4. Consequence of lexical translation #1: phonetic arbitrariness, unnaturalness
   Arbitrariness is an obvious and undisputed property of Vocabulary Insertion (VI), i.e. the translation of morphemic information when spell-out transforms morpho-syntactic structure into phonological units. It follows from the lexical character of VI, which is just as obvious and without alternative: related to their morpho-syntactic correlates, phonological forms are drawn from the mental lexicon. The lexical character of VI and its ensuing arbitrariness is never argued for because it goes without saying. Hence 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.

If intermodular communication is lexical, 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). Hence given phonetic arbitrariness, what phonologists call [labial] etc. is not labial in any way in the phonology. The use of [labial] etc. is shorthand for a colourless phonological prime α whose phonetic correlate is labial, but which could have any other phonetic correlate as well. In the phonology, α is not labial in any way and phonology does not "know" about its phonetic correlate.

It follows that naturalness is not present in or created by the phonology: n → ŋ /__k is just as phonologically legitimate as, say, n → ŋ / __p or r → ŋ / __i. This is the programme of substance-free phonology: any object or process and their reverse are legal, phonology does not evaluate what it is doing. That is, the reason why the vast majority of phonological processes are "natural" lies outside of phonology: for example the fact that phonological processes are the grammaticalized versions of phonetic precursors, which are natural.

5. Consequence of lexical translation #2: no diacritics

   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.

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

6. Consequence of lexical translation #3: only the lower part of phonology is arbitrary

   It follows from the architecture described that only those items that are translated into one another, i.e. which make an interface-dictionary entry α ↔ Y, entertain an arbitrary relationship. For phonology this means that only segmental (or melodic) items (distinctive features) are concerned: in a regular autosegmental representation, they occur below the skeleton and have phonetic correlates ([labial] is related to labiality, etc.). By contrast, items at and above the skeleton are not concerned since they have no phonetic correlate: a nucleus, a grid mark or a prosodic word has no phonetic value associated. Hence unlike features, these objects and their computation are not arbitrary.

   It is argued that this distinction between items below and at/above the skeleton is indicative of the existence of distinct modules within the phonology.

7. How modules are related to the real world (external to the body)

   Intermodular interfaces relate cognitive systems. Hence they are incompetent when it comes to the relationship between cognitive systems and the outside world, i.e. items that occur beyond the body such as light waves, odours, the acoustic signal etc. The

relationship between mental and real-world items is made of arbitrary associations: chunks of a real-world continuum are associated to discrete mental categories. In phonology, relevant mental categories are phonemes, which are associated to phonetic correlates. Sound is thus categorized into phonemes just like, say, a certain band of wave lengths is into colours.

8. Module-specific implementation of hierarchy: trees vs. lateral relations
   There is broad agreement that language is hierarchical in nature. But are hierarchical relations expressed in the same way in all areas of language? Are trees the only way to express hierarchy? It is argued that there are at least two different ways to express hierarchical relations: by trees and by lateral relations. The former occur in morpho-syntax, the latter in phonology, and both are in complementary distribution: in presence of one, the other is absent. Hence phonology is "flat": there are no trees in this module. Lateral relations – government and licensing – root in Dependency Phonology and are the genuine contribution of Government Phonology to the field.
   The existence of different means to express hierarchical relations, it is argued, has two reasons. It is a design property of morpho-syntax to carry out concatenation of pieces stored in long term memory. There is no concatenation in phonology, though: (in production) phonology only interprets whatever was concatenated prior to its activity. In morpho-syntax, tree-type hierarchy is created by concatenation (the minimalist device Merge), and by nothing else. Hence in absence of concatenation in phonology, there could not be any trees: no merge, no trees. This explains the absence of recursion in phonology, a long-standing observation: no trees, no recursion.
   On the other hand, the input to phonology is a linearized string, while there is no linearity in morpho-syntax. It is argued that design properties (concatenation present vs. absent) and input conditions (linearity present vs. absent) determine the kind of hierarchy found: morpho-syntax is driven into trees because of concatenation and could not implement lateral relations since it lacks linearity. Phonology could not work with trees because it does not concatenate anything; its linear input leads to lateral relations.