

On the Nature of the Input in Optimality Theory

Fabian Heck, Gereon Müller, Ralf Vogel,
Silke Fischer, Sten Vikner, Tanja Schmid

Abstract:

The input has two main functions in optimality theory (Prince and Smolensky (1993)). First, the input defines the candidate set, in other words it determines which output candidates compete for optimality, and which do not. Second, the input is referred to by faithfulness constraints that prohibit output candidates from deviating from specifications in the input. Whereas there is general agreement concerning the relevance of the input in phonology, the nature of the input in syntax is notoriously unclear. In this article, we show that the input should not be taken to define syntactic candidate sets, and that all syntactic faithfulness constraints can straightforwardly be reformulated as constraints on outputs. In view of this, we contend that the input can be completely dispensed with in syntax, in contrast to what is the case in phonology; and we argue that this asymmetry between phonology and syntax is due to a basic, irreducible difference between these two components of grammar: Syntax is an information preserving system, phonology is not.

1. Introduction

In standard optimality theory (see Prince and Smolensky 1993), a generator function GEN maps an input I to a candidate set C of competing structures. C is then evaluated with respect to an ordered list of constraints. The structure with the best constraint profile is the grammatical output. In this model, the input I has two basic functions. First, it defines the candidate set. In other words, only structures that emerge from the same input are possible competitors. Second, its existence is presupposed by faithfulness constraints, that is to say, constraints that penalize deviation of an output candidate from the specifications given in the input.

In optimality theoretic phonology, it is usually assumed that the input is an underlying representation (henceforth UR): a (possibly abstract) lexical item. In contrast, it is still an open question what the input consists of in optimality theoretic syntax. In this article we consider the notion of input in optimality theory in more detail. The result is as radical as it is simple: We contend that the concept of input can be dispensed with in optimality theoretic syntax.

We will proceed as follows. In section 2. we review the role that inputs play in optimality theoretic phonology and conclude that there is good reason to assume inputs in this domain. On this basis we turn to the role of inputs in syntax in sections 3. and 4. In section 3. we argue that inputs should not be taken to define candidate sets in syntax. In section 4. we show that syntactic faithfulness

constraints can always be straightforwardly reformulated as constraints that do not rely on the notion of input; hence, we conclude that there may in fact be no reason to adopt inputs in syntax at all and that this concept can simply be dispensed with.¹ Finally, section 5. is concerned with the question of why phonology and syntax should differ in this way. We suggest that this is due to a basic, irreducible difference between phonology and syntax: The latter is information preserving, the former is not.

2. Inputs in Phonology

In generative phonological theory, and particularly in derivational phonology, output representations are standardly assumed not to contain abstract material like, for instance, traces in syntactic theory. However, derivational phonology relies on rules which are not structure preserving; in other words, information is lost during the derivation. As a consequence, the input (i.e., the UR) and every intermediate derivational level are necessary in order to keep the derivation transparent.

2.1. Phonological Systems without Abstract Material

The study of Lardil phonology in Kenstowicz and Kisseberth (1979: 109-115) may serve to demonstrate this point.

2.1.1. *Feeding in Lardil (Kenstowicz and Kisseberth 1979)* Two rules and their interaction are considered: Apocope, a rule which deletes the final vowel in a trisyllabic structure, and the rule of non-apical deletion (NAD), which deletes a non-apical word-final consonant:

- (1) Apocope

$$V \rightarrow \emptyset / VCVC _ \#$$
- (2) NAD

$$C_{[-apical]} \rightarrow \emptyset / _ \#$$

The effects of the rules are exemplified in (3) and (4), respectively:

- (3) Apocope
 - a. Input: /mayara/ ‘rainbow’
 - b. mayara (Apocope) →
 - c. mayar

¹We hasten to add that this conclusion is not valid for all kinds of optimality theoretic syntax (see footnote 32).

- (4) Non-Apical Deletion (NAD)
- a. Input: /ŋaluk/ ‘story’
 - b. ŋaluk (NAD) →
 - c. ŋalu

The interaction of Apocope and NAD is shown in (5). First, Apocope applies to the UR /wulunka/. This creates [wulunk], a structure with the appropriate phonological context for NAD to apply. In other words, Apocope feeds NAD.

- (5) Interaction of Apocope and NAD
- a. Input: /wulunka/ ‘fruit’
 - b. wulunka (Apocope) →
 - c. wulunk (NAD) →
 - d. wulun

The feeding relation is encoded by successive rule application that creates intermediate levels of representation. The output [wulun] on its own does not encode the history of the derivation. All intermediate levels have to be considered.

The next section shows how feeding relations are reconstructed in the optimality theoretic analysis of Prince and Smolensky (1993: 97-125).²

2.1.2. Feeding Interaction in Optimality Theory (Prince and Smolensky 1993)

Standard optimality theoretic phonology rejects the existence of intermediate levels; mapping applies directly from the input onto the output. One abstract level (the input) is enough to model feeding interactions within an optimality theoretic approach. The two rules are translated as output constraints:³

- (6) FREE-V
Word-final vowels must not be parsed.⁴
- (7) CODA CONDITION
A coda consonant can have only apical place [. . .].⁵

²It should be noted that the presentation given here is simplified and just presents the basic gist of the analysis given in Prince and Smolensky (1993).

³Prince and Smolensky’s analysis is based on the assumption that what is at issue is not language-specific rules but the interaction of very general constraints about the prosodic shape of words. Therefore, the constraints look somewhat different from the rules. However, basically FREE-V parallels Apocope, and CODA CONDITION parallels NAD.

⁴Note that this constraint must be overruled by a constraint that prohibits the truncation in minimal words, cf. Prince and Smolensky (1993: 101). This mimics the effects of deleting only vowels that are preceded by at least two other vowels; see the Apocope rule in (1).

⁵In Prince and Smolensky (1993) the feature “coronal” is used. To facilitate comparison, we use Kenstowicz and Kisseberth’s (1979) feature “apical” here.

If a faithfulness constraint (PARSE in this case) is added, the feeding interaction can be implemented without intermediate levels and successive rule application:

- (8) PARSE (Prince and Smolensky 1993: 85)
Underlying segments must be parsed into syllable structure.

GEN provides a very large candidate set, which also contains the actual winner under the ranking FREE-V, CODA CONDITION \gg PARSE. The evaluation of (a subset of) these candidates is shown in tableau T₁.

T₁: *Feeding Interaction in Lardil*

Input: /wulunka/	FREE-V	CODA COND	PARSE
O ₁ : wulunka	*!		
O ₂ : wulunk		*!	*
☞ O ₃ : wulun			**
O ₄ : wulu	*!		***
O ₅ : wul			***!*

deviate too much from the input are excluded by the faithfulness constraint PARSE, which penalizes deletion (see output candidate O₅).⁶ Note that the input is crucial for this analysis, since a violation of faithfulness (i.e., deletion) can only be defined *relative* to the input. Only the comparison between input and output makes clear how often PARSE is actually violated by an output structure.

The optimality theoretic reconstruction of feeding interactions suggests that it is possible to do away with intermediate levels. The only representations left are input and output. The question arises whether the most radical option is available as well: a system that exclusively deals with output structures.

2.2. *Phonological Systems with Abstract Material*

In a system that only operates on output representations, information that was located at other levels of representation before (including the input) must be encoded in the output structure. There have been proposals within phonological theory that rely on more abstract output representations. The question is this: Can these proposals dispense with the input?

⁶Actually, output candidate O₅ is independently blocked by the requirement on minimal words in Lardil (cf. footnote 4). There is, however, independent evidence for PARSE provided by deletion applying to words with more syllables.

2.2.1. *The Parse/Fill System* A system with enriched output structures is provided by the theory of syllabification given in Prince and Smolensky (1993: 85-96). Central to the system is an assumption which has become known as *containment*: “Phonologically deleted segments are present in the output but unparsed [. . .] The IO-Faithfulness constraint PARSE regulates this mode of deletion” (McCarthy and Prince 1995b: 8). Deleted elements – included between the diacritics “⟨” and “⟩” – are present in the output structure, but invisible for phonetic interpretation. The interaction between such elements and the prosodic constraint in (9), which penalizes the presence of codas, is shown in tableau T₂:

- (9) NOCODA
Syllables must not have a coda.

T₂: *Containment and PARSE*

Input: /CVC/	PARSE	NOCODA
☞ O ₁ : .CVC.		*
O ₂ : .CV.⟨C⟩	*!	
O ₃ : ⟨CVC⟩	*!***	

The PARSE-violations of the output candidates O₂, O₃ can be calculated without reference to the input. All the necessary information is provided by the diacritics that mark deleted elements.

The same point can be made with an operation that is the opposite of deletion: epenthesis. This time, the information about which element was epenthesized is encoded by the diacritic “□”. The “□” is a kind of variable that opens a slot in the prosodic structure, which is not yet filled in phonology. Epenthesis violates the faithfulness constraint FILL:

- (10) FILL
Syllable positions must be filled with underlying segments.

Again, all the output candidates in tableau T₃ have their violations of FILL directly encoded by the diacritic “□”. No reference to the input is necessary.

T₃: *Containment and FILL*

Input: /CVC/	FILL	NOCODA
☞ O ₁ : .CVC.		*
O ₂ : .CV□.C□.	*!*	
O ₃ : .CV.C□.	*!	

Since the relevant information is now encoded twice, in the candidates themselves and in the input (comparison between input and output makes transparent which elements were deleted and which were epenthesized), it seems that one could do away with the input and think of PARSE/FILL not as faithfulness constraints, but as constraints on outputs that penalize the presence of “□” and “⟨. . .⟩” in the output (see also the discussion in section 4.).

2.2.2. *Syllabification in Lardil (Prince and Smolensky 1993)* Let us see what the PARSE/FILL-analysis of Lardil syllabification in Prince and Smolensky (1993: 97-125) (see section 2.1.2.) looks like. We return to the interaction between Apocope and non-apical deletion (NAD):

- (11) Interaction of Apocope and NAD
- a. Input: /wulunka/ ‘fruit’
 - b. wulunka (Apocope) →
 - c. wulunk (NAD) →
 - d. wulun

As discussed before, Apocope feeds NAD, which can easily be accounted for in a derivational approach by ordering Apocope before NAD. However, in the optimality theoretic PARSE/FILL-analysis elements deleted by Apocope are still “visible” in the output in some sense; see tableau T₄.

T₄: *Feeding Interaction in Lardil with PARSE/FILL*

Input: /wulunka/	FREE-V	CODA COND	PARSE
O ₁ : wulunka	*!		
O ₂ : wulunk⟨a⟩		*!	*
☞ O ₃ : wulun⟨ka⟩			**
O ₄ : wulu⟨nka⟩			*!***
O ₅ : wul⟨unka⟩			*!****

all the information about which elements have been deleted and which elements have been parsed is represented in the output candidates and is therefore

available without recourse to the input.^{7,8}

Evidence for the PARSE/FILL-system might serve as an argument for the assumption that phonological representations are more abstract than traditionally assumed. As a side effect this could lead to the abolition of the input. However, there are some arguments within optimality theory against the claim that phonology is abstract to such a degree. As a consequence, these are also arguments against an input-free system.

2.3. *Against Abstract Representations in Phonology*

In this section we recapitulate some arguments that were given in the literature against abstract representations in phonological theory and we will present a new argument based on the concept of input optimization in favor of the presence of input in phonology.

2.3.1. *Correspondence Theory* In McCarthy and Prince (1995a), the PARSE/FILL system is replaced by Correspondence Theory (CT). Correspondence Theory rejects the assumption of containment: In CT, both input and output consist entirely of overt non-abstract phonological material. Additionally, there is a relation between the input segments and the output segments, the so-called correspondence. In (12) this relation is expressed by the indices on the segments:

$$(12) \quad \begin{array}{l} \text{input:} \quad s_1 s_2 s_3 s_4 \\ \text{output:} \quad s'_1 s'_2 s'_3 s'_4 \end{array}$$

Segments with the same index are in a correspondence relation. Deletion and epenthesis can then be read off the lack of a corresponding element. PARSE translates into MAX-input/output (MAX-IO), and FILL translates into DEP-input/output (DEP-IO) (see McCarthy and Prince 1995a: 264; MAX stands for “maximality,” DEP stands for “dependence”):

⁷Note that the purpose of tableau T₄ is not to provide motivation for containment. Evidence for the claim that unparsed elements still have some impact is provided by optimal outputs where FREE-V cannot be violated despite what is suggested at first sight; for instance, /puʈuka/ → puʈu, /muŋkumuŋku/ → muŋkumu; see Kenstowicz and Kisseberth (1979: 315), Prince and Smolensky (1993: 101). However, for reasons of coherence we stick to our example /wulunka/, since it is sufficient to demonstrate how information formerly encoded in the input is now available in the output.

⁸One may ask why the deleted [a] of O₂ does not count as visible; in this case it should block the violation of CODA COND. The answer is that CODA COND is defined with respect to syllables. Under the assumption that unparsed elements can never have any effect on syllabification, the unparsed [a] in O₂ is irrelevant for CODA COND. Moreover, as remarked by an anonymous TLR-reviewer, a deleted vowel (as [a] in O₂) could never be visible for FREE-V, since this would undermine the basic purpose of FREE-V; see Prince and Smolensky (1993: 101).

- (13) a. MAX-input/output
Every segment of the input has a correspondent in the output.
b. DEP-input/output
Every segment of the output has a correspondent in the input.

For the Lardil example in 2.1.2., the correspondence-based analysis yields tableau T_5 .

T_5 : *Feeding Interaction in Lardil with Correspondence*

Input: /w ₁ u ₂ l ₃ u ₄ n ₅ k ₆ a ₇ /	FREE-V	CODA COND	MAX-IO
O ₁ : w ₁ u ₂ l ₃ u ₄ n ₅ k ₆ a ₇	*!		
O ₂ : w ₁ u ₂ l ₃ u ₄ n ₅ k ₆		*!	*
☞ O ₃ : w ₁ u ₂ l ₃ u ₄ n ₅			**
O ₄ : w ₁ u ₂ l ₃ u ₄	*!		***
O ₅ : w ₁ u ₂ l ₃			***!*

Among the arguments for preferring CT over the PARSE/FILL model are the following two: First, the variable “□” in the system PARSE/FILL provides only a prosodic position. The question of how this position is to be interpreted phonetically was thought to be a matter of post-phonological processes. But this assumption raises a conceptual problem (see McCarthy and Prince 1995a: 265; McCarthy and Prince 1995b: 8), because the choice of the epenthetic element follows phonological criteria. Therefore, the phenomenon of epenthesis should be treated within the phonological component and not post-phonologically. Second, CT can be generalized from input/output correspondence to base/reduplicant correspondence (McCarthy and Prince 1995a,b) or output/output correspondence (Benua 1997); see the discussion in McCarthy and Prince (1995b chapter 2.3). It is not clear how to extend the PARSE/FILL approach to cover these phenomena.

Arguably, the CT approach has become the standard view by now. The important point in the present context is that CT rejects abstract outputs and strengthens the notion of input.

2.3.2. *Input Optimization* Another argument for the relevance of the notion of input in phonology may be constructed from what has become known as input or lexicon optimization (Prince and Smolensky 1993: 192):

- (14) Input Optimization
Suppose that several different inputs I_1, I_2, \dots, I_n when parsed by a grammar G lead to corresponding outputs O_1, O_2, \dots, O_n , all of which are realized as the same phonetic form Φ – these inputs are all *phonetically equivalent* with respect to G . Now one of the outputs must be the most

harmonic, by virtue of incurring the least significant violation marks: Suppose this optimal one is labeled O_k . Then the learner should choose, as the underlying form for Φ , the input I_k .

Let us briefly consider an example. In the history of English a change took place from [sk] to [ʃ] (see *shadow*, *shape*, *ship*, *shoulder*, etc.), as shown in (15).⁹

(15) Old English *scip* [skɪp] \longrightarrow [sɪp] \longrightarrow English *ship* [ʃɪp]

Some Germanic languages did not undergo this change, so that the old [sk] is still visible there; compare Old Norse *skip* [sk], Icelandic *skip* [sk], Danish *skib* [sk].

At some point in the development of English, there must have been a new ranking of constraints that led a generation Γ_1 of speakers to transform the [sk] output of their parents to an unfaithful [ʃ]-output. Suppose for the sake of argument that the change concerned the demotion of the pertinent MAX-IO faithfulness constraint below a constraint that prohibits [sk] clusters in the onset, and that we may refer to as *SK-ON for present purposes. According to input optimization, this [ʃ]-output then led a later generation Γ_n to an UR with [ʃ], i.e., an input that the optimal output is faithful to. What was pronounced as [skɪp] at least until the year 600 has been pronounced as [ʃɪp] at least since the year 1100. Hence, we can conclude that around 1100, the only available input for [ʃɪp] was [ʃ]; the original [sk]-input has vanished, due to input optimization. Interestingly, foreign [sk]-words that entered the language after this period (like, e.g., [skɪp], *skip*) are not realized with an initial [ʃ], but show up with an initial [sk]; this suggests that their presence in English was sufficient to trigger a parsing-induced demotion of *SK-ON again (see Tesar and Smolensky (2000)). Crucially, if the original [skɪp] were still an available input at this point, we would wrongly expect *ship* to be realizable as [skɪp] again. The fact that it is not can therefore be taken as an argument that input optimization did indeed occur, with [skɪp] not being a possible input for *ship* anymore. Nowadays, forms like [skɪp] (*skip*) and [ʃɪp] (*ship*) are both faithful outputs. Thus, we have an argument for input optimization and, a fortiori, for the existence of inputs.

To conclude this section, inputs are indispensable for the definition of faithfulness constraints in phonology if output structures are to be kept relatively simple.¹⁰ In a more abstract phonological system like the PARSE/FILL system, the input might be dispensable, as far as faithfulness is concerned. However, this system was abandoned for independent reasons. Furthermore, there is an argument in

⁹According to Pinsker (1959: 85) this change had been completed before the year 1100; according to Ross (1965: 109/121) before the year 1000.

¹⁰Note that this conclusion is not affected by the existence of proposals in the phonological literature that pursue the elimination of *abstract* inputs, i.e., URs. See Burzio (1996) and Hayes (1999).

favor of the concept of input that relies on the notion of input optimization. With this in mind, let us now turn to syntax. We begin with the role of inputs in the definition of syntactic candidate sets.

3. Inputs in Syntax: Candidate Sets

In this section, we argue that none of the existing notions of input in optimality theoretic syntax is suited to fulfill one of the two major tasks that are standardly attributed to it: The input is usually taken to define the candidate set; only candidates that emerge from the same input can be in the same candidate set and can compete for optimality.¹¹

Different proposals as to what the input looks like have been made in the literature. By and large we can classify the proposals as the concepts of *non-structured* input (compare the concept of “numeration” in Chomsky 1995), *partially structured* input (see Grimshaw 1997a), and *highly structured* input (see Legendre, Smolensky and Wilson 1998, Vikner 1997, Pesetsky 1998). We address these concepts in turn.

3.1. *Non-structured Inputs*

If the input is non-structured, the following scenario is in principle possible: Two outputs, both grammatical but otherwise radically different, emerge from one and the same input, i.e., from the same candidate set. This scenario is not desirable for two reasons, one conceptual and one empirical: First, intuitively, two candidates should only compete if they are sufficiently similar, e.g., with respect to their semantic interpretation. Second, if two significantly different candidates compete, it is notoriously difficult to ensure that both can be optimal. Often it is extremely unlikely that they can have an identical constraint profile.¹²

3.1.1. The Conceptual Problem: Unwanted Competition Let us look at a concrete example (following Archangeli and Langendoen 1997: 213-215). A totally non-structured input like (16-a) gives rise to the two structures (16-b) and (16-c):

¹¹Whereas the notion of “candidate set” is predominant in optimality theoretic literature, the notion of “reference set” is primarily used in work that centers around competition-based versions of the minimalist program (see Chomsky 1995). At least for the purposes of the present article, the notions can be considered to be equivalent.

¹²Throughout, we ignore the possibility of constraint ties. Constraint ties may in some cases be an appropriate means to derive multiple optimality in a single candidate set, but they can only be successfully employed when the candidates in question are otherwise sufficiently similar.

- (16) a. Input:
 {that, likes, John, Mary}
 b. . . . that [_{VP} John [_{V'} likes Mary]]
 c. . . . that [_{VP} Mary [_{V'} likes John]]

Clearly, (16-b) and (16-c) do not mean the same thing; therefore, they should not be in the same competition. A first solution to this problem could be to assume that lexical items bear case markers. Then the optimal outputs (16-b) and (16-c) will emerge from different inputs (and hence, different competitions) with different case markings. One of those inputs is given in (17-a):

- (17) a. Input:
 {that, likes, John_{acc}, Mary_{nom}}
 b. * . . . that [_{VP} John_{acc} [_{V'} likes Mary_{nom}]]
 c. . . . that [_{VP} Mary_{nom} [_{V'} likes John_{acc}]]

The output candidate (17-b) that implements the word order of (16-b) by mapping the accusative NP into SpecV, and the nominative NP into the complement position of V, will never be optimal, given the input in (17-a). An optimal output that exhibits this word order has to emerge from another input, one with different case markers.

However, the examples in (18), which involve embedding, show that the input in (17) is not yet structured enough to solve the problem:

- (18) a. Input:
 {thinks, that, likes, John_{acc}, Mary_{nom}, Carl_{nom}}
 b. Carl_{nom} thinks [_{CP} that [_{VP} Mary_{nom} [_{V'} likes John_{acc}]]]
 c. Mary_{nom} thinks [_{CP} that [_{VP} Carl_{nom} [_{V'} likes John_{acc}]]]

Both versions in (18) are optimal and emerge from the same input. But they do not mean the same thing. Therefore, we are left with the same problem as before. In the next step one could try to add features like [matrix] and [embedded] to the input:

- (19) a. Input:
 {thinks, that, likes, John_{acc}, Mary_{nom,matrix}, Carl_{nom,embedded}}
 b. *Carl_{nom,embedded} thinks [_{CP} that [_{VP} Mary_{nom,matrix} [_{V'} likes John_{acc}]]]
 c. Mary_{nom,matrix} thinks [_{CP} that [_{VP} Carl_{nom,embedded} [_{V'} likes John_{acc}]]]

Again, (19-b) cannot be an optimal output for the given input for obvious reasons. The string must be the optimal output of another competition. But the input is still not rich enough. If the structures get more complicated, the problem emerges again:

- (20) a. Anna_{nom,matrix} thinks [CP that Carl_{nom,embedded} thinks
[CP that [VP Mary_{nom,embedded} [v' likes John_{acc}]]]]
b. Anna_{nom,matrix} thinks [CP that Mary_{nom,embedded} thinks
[CP that [VP Carl_{nom,embedded} [v' likes John_{acc}]]]]

This reasoning suggests that the input has to be enriched even further: it has to be structured.

3.1.2. *Different Constraint Profiles: The Case of Binding* Another problem for non-structured inputs involves binding in German. The examples in (21) show that the reciprocal *einander* ('each other') can be bound either by the subject *die Gastgeber* ('the hosts', see (21-b)) or the direct object *die Gäste* ('the guests', see (21-c)).

- (21) a. Input:
{dass, vorstellten, die_{nom}, Gastgeber_{nom}, die_{acc}, Gäste_{acc}, einander_{dat}}
b. ... dass [VP die Gastgeber₁ [v' die Gäste₂ [v' einander₁
that the hosts the guests each other
vorstellten]]]
introduced
'that each of the hosts introduced the guests to the other hosts'
c. ... dass [VP die Gastgeber₁ [v' die Gäste₂ [v' einander₂
that the hosts the guests each other
vorstellten]]]
introduced
'that the hosts introduced each of the guests to the other guests'

It is highly improbable that the two variants can have the same constraint profile. For instance, a minimal distance condition would favor (21-c), whereas a subject-as-binder condition would favor (21-b). Since (21-b,c) are both grammatical, this in turn means that they have to emerge from different competitions, i.e., different inputs. A possible solution is the assumption that binding indices are already present in the input:

- (22) a. Input:
{dass, vorstellten, die_{nom}, Gastgeber_{nom,1}, die_{acc}, Gäste_{acc,2}, einander₂}
b. *... dass [VP die Gastgeber₁ [v' die Gäste₂ [v' einander₁
that the hosts the guests each other
vorstellten]]]
introduced
'that each of the hosts introduced the guests to the other hosts'

- c. ... dass [_{VP} die Gastgeber₁ [_{V'} die Gäste₂ [_{VP} einander₂
that the hosts the guests each other
vorstellten]]]
introduced
‘that the hosts introduced each of the guests to the other guests’

A candidate like (22-b) then either fatally violates faithfulness because it does not respect the binding relations given in the input, or (if GEN is not allowed to manipulate binding indices of the input) it is not part of the same competition as (22-c) in the first place.

3.1.3. *Different Constraint Profiles: The Case of Wh-movement* The next empirical problem for non-structured inputs involves *wh*-movement. The example in (23-b) involves long *wh*-movement out of an embedded clause, (23-c) involves short *wh*-movement within an embedded clause. The two versions involve the same lexical material.¹³

- (23) a. Input:
{*sagte*, *mag*, *Fritz*_{nom,matrix}, *Maria*_{nom,emb.}, *wen*_{acc}, *dass*_[+wh], *dass*_[-wh]}
- b. (Ich weiß) [_{CP} *wen*₁ *dass*_[+wh] *Fritz* *sagte* [_{CP} *t*₁ *dass*_[-wh] *Maria* *t*₁
(I know) who that Fritz said that Maria
mag]]
likes
‘(I know) who Fritz said Maria likes’
- c. (Ich weiß) [_{CP} *dass*_[-wh] *Fritz* *sagte* [_{CP} *wen*₁ *dass*_[+wh] *Maria* *t*₁
(I know) that Fritz said who that Maria
mag]]
likes
‘(I know) that Fritz said who Maria likes’

It is extremely unlikely that (23-b) and (23-c) can have the same constraint profile: The two candidates differ with respect to the number of steps involved in *wh*-movement and with respect to the length of the paths of *wh*-movement. But both are well formed. Hence, it has to be ensured that they do not compete.

One could try to encode in the input whether a verb that in principle allows both [+wh]- and [-wh]-complements actually takes one or the other. This would clearly distinguish (23-b) (*sagte* selects [-wh]) from (23-c) (*sagte* selects [+wh]).

¹³Both versions have two complementizers (*dass*_[+wh], *dass*_[-wh]), one in the embedded clause, and one in the matrix clause. A co-occurrence of a *wh*-phrase in SpecC and a complementizer *dass* in C is possible in some (colloquial) varieties of German that allow violations of the doubly-filled Comp filter; it is only these varieties that we are currently interested in.

But this solution still runs into problems with multiple questions of the type discussed by Baker (1970):

- (24) a. Input:
 {wonders_[+wh], bought, who_{nom,matrix}, we_{nom,embedded}, where, what_{acc}}
 b. Who₁ Q₃ t₁ wonders [_{CP} where₂ we bought what₃ t₂] ?
 c. Who₁ t₁ wonders [_{CP} where₂ Q₃ we bought what₃ t₂] ?

<Q₃,what₃> in (24-b,c) is an output chain.¹⁴ The surface string in (24-b,c) is ambiguous, and the scope marker Q₃ encodes different possible scope positions for the coindexed *wh*-phrase. The different meanings are given in (25-a,b), respectively:

- (25) a. For which person *x* and which thing *y*: *x* wonders where we bought
 y
 b. For which person *x*: *x* wonders where we bought what

An appropriate answer to the question in (24-b) would be a sentence like *John wonders where we bought the car, and Mary wonders where we bought the beer*, whereas an appropriate answer to (24-c) would be something like *John wonders where we bought what*.

Again, the point is that the two variants (24-b,c) represent different readings, and it is highly improbable that they can have the same constraint profile (again, they differ at least with respect to the length of the chain link). There is no way, however, to encode this in the input by adding a feature to the matrix verb.

3.2. Partially and Highly Structured Inputs

The conclusion of the previous section is the following: If the input defines which candidates are in competition, then it has to be more complex. There have been different proposals in the literature about how complex the input should be.

3.2.1. *Partially Structured Input (Grimshaw 1997a)* Grimshaw (1997a) assumes that the input consists of a predicate-argument structure, together with some specifications about tense and aspect. Functional categories are never part of the input, but can be freely added by GEN.

- (26) a. Input:
 ((likes, John), Mary)
 b. *... that [_{VP} John [_{V'} likes Mary]]

¹⁴Alternatively, one can think of this chain as a copy chain whose base is spelled out or as a traditional movement chain in an output-LF.

c. . . . that [_{VP} Mary [_{V'} likes John]]

If two candidates can only compete if they realize the same predicate-argument structure, then (26-b,c) do not compete. Only (26-c) is a well-formed candidate with respect to the given input. However, this is not yet sufficient to avoid the problems discussed in subsections 3.1.2./3.1.3.

Grimshaw (1997a: 376) therefore proposes that competing candidates must have the same meaning, in a sense to be specified: “Competing candidates have non-distinct logical forms, in a sense which must be made precise by further research, but which certainly must entail that they are truth-functionally equivalent.” There are basically two strategies how this statement can be spelled out: First, the LF is not part of the input. Only those candidates that have been generated from the same input and that additionally result in non-distinct LFs are in the same competition. Second, the LF-structure is already encoded in the input.¹⁵

If the first strategy is chosen, the concept of input is weakened: The notion of candidate set must then be defined at least partially with respect to properties that are not located in the input. Conceptually, this is unattractive. Since, as we will argue, the input turns out to be redundant in general, parsimony then suggests an input-free definition of candidate sets.¹⁶

The second strategy, on the other hand, seems very close to what has been proposed by Legendre, Smolensky and Wilson (1998); this approach will be discussed in the next section. However, it turns out that the second strategy is not what Grimshaw had in mind; otherwise, the following problem would arise: If the input contained a highly specified LF, there would be unfaithful output candidates that deviate from the input in some meaning-related aspect. But then it would no longer necessarily be the case that competing candidates have non-distinct LFs, clearly against Grimshaw’s explicit assumption.

3.2.2. *Highly Structured Input (Legendre, Smolensky and Wilson 1998)* Legendre, Smolensky and Wilson (1998) assume that each candidate set is fully determined by the input. They also assume that the input contains a “target LF”

¹⁵See Grimshaw (1997a: 376): “It may turn out that the input should include a specification of LF-related properties, such as scope.”

¹⁶The question of what the input-independent definition should look like is a very delicate matter, and we will not try to give a comprehensive answer here. Many different proposals have been made in the literature (see in particular the discussion in Sternefeld 1997). Tentatively one might think of something like (i):

- (i) *Candidate set*
Two candidates O and O' are in the same candidate set if and only if they have the same LF.

(their so-called “index”). Basically, “target LF” means that the structure contains information about the target scope positions of *wh*-phrases. This means that the input is a highly articulated structure which encodes an interpretation (but not necessarily the actual one). In contrast to what is the case in Grimshaw’s system, it is then possible that not all competing candidates have the same LF-output, although they all emerge from the same target LF, because there can be unfaithful candidates that deviate from the target LF.

The basic idea of the system can be demonstrated with some examples of possible interpretations of *wh*-questions in Chinese: If an adjunct *wh*-phrase is located within an embedded declarative clause, an interpretation with wide scope of the *wh*-phrase is possible (cf. (27)). If the *wh*-phrase is located within an embedded question, only the interpretation of an embedded multiple question is available (cf. (28)): the *wh*-phrase must have narrow scope, i.e., embedded questions are islands for *wh*-extraction of adjuncts.

- (27) a. Ni renwei [_{CP} Lisi yinggai zenmeyang chuli zhe-jian shi] ?
 you think Lisi should how treat this-CL affair
 b. ‘How₁ do you think Lisi should treat this affair t₁?’
- (28) a. Ni xiang-zhidao [_{CP} shei zenmeyang chuli zhe-jian shi] ?
 you ask yourself who how treat this-CL affair
 b. ‘*How₁ do you wonder who has treated this affair t₁?’
 c. ‘You wonder who treated this affair how.’

Since Chinese is a *wh*-in-situ language, extraction cannot be observed overtly. Legendre, Smolensky and Wilson (1998) assume that chain formation has already taken place at S-structure: An abstract scope marker binds the *wh*-item from the scope position. The following constraints are adopted:

- (29) a. SELECT
 Lexical selection has to be observed.
 b. BARRIER^{2[-ref]} (BAR^{2[-ref]})
 A non-referential (adjunct) chain link must not cross two barriers.
 c. PARSE SCOPE
 Scope that is marked in the input has to be realized by chain formation in the output.

SELECT is violated if, for instance, a verb that selects a [+wh]-complement in the input has a [-wh]-complement in the output. BAR^{2[-ref]} penalizes chains whose head is an adjunct and that cross two barriers; the assumption is that every maximal projection that is not L-marked counts as a barrier. IP and VP are the typical cases. Finally, PARSE SCOPE is violated if the scope of a *wh*-phrase which is indicated in the input is not realized in the output by chain formation. The Chinese

ranking is $\text{SELECT} \gg \text{BAR}^{2[-ref]} \gg \text{PARSE SCOPE}$.

Tableau T_6 shows what happens in the context of *wh*-islands: Wide scope in the input (marked by the Q-operator) is reduced to narrow scope in the output in order to avoid the crossing of matrix VP and IP, in violation of PARSE SCOPE.

T_6 : *Wh-Islands for Wide Adjunct-Scope in Chinese*

Input: $Q_1 \dots V_{[+wh]} [CP t'_1 \dots wh_1 \dots]$	SELECT	$\text{BAR}^{2[-ref]}$	PARSE SCOPE
$O_1: Q_1 \dots V_{[+wh]} [CP t'_1 \dots wh_1 \dots]$		*!	
$\leftarrow O_2: - \dots V_{[+wh]} [CP Q_1 \dots wh_1 \dots]$			*

If narrow scope is in the input, both PARSE SCOPE and $\text{BAR}^{2[-ref]}$ can be satisfied without scope reduction; compare tableau T_7 .

T_7 : *Narrow Adjunct-Scope in Chinese*

Input: $- \dots V_{[+wh]} [CP Q_1 \dots wh_1 \dots]$	SELECT	$\text{BAR}^{2[-ref]}$	PARSE SCOPE
$\leftarrow O_1: - \dots V_{[+wh]} [CP Q_1 \dots wh_1 \dots]$			
$O_2: Q_1 \dots V_{[+wh]} [CP - \dots wh_1 \dots]$		*!	*

Thus, the difference in the inputs of T_6 and T_7 is “neutralized” to one and the same output. Something surprising happens in the context of embedded declaratives: Here the operator with wide scope violates $\text{BAR}^{2[-ref]}$ as in tableau T_6 . But this time scope reduction is impossible since this would imply turning the embedded declarative clause into an embedded question in violation of the high-ranked SELECT; therefore, the wide scope output emerges as optimal. This is shown in tableau T_8 .

T_8 : *Declaratives and Wide Adjunct-Scope in Chinese*

Input: $Q_1 \dots V_{[-wh]} [CP \dots wh_1 \dots]$	SELECT	$\text{BAR}^{2[-ref]}$	PARSE SCOPE
$\leftarrow O_1: Q_1 \dots V_{[-wh]} [CP \dots wh_1 \dots]$		*	
$O_2: - \dots V_{[-wh]} [CP Q_1 \dots wh_1 \dots]$	*!		*

To conclude, this approach chooses the second of the two strategies discussed in the last subsection. It relies on a highly articulated LF in the input. This enables the approach to use the technique of neutralization to solve the problem of total ungrammaticality of extraction out of *wh*-islands. However, from the point of view of the input discussion, this clearly raises a conceptual problem: Where does the highly structured input come from? It must have been created by another

generator, say Σ .¹⁷ But so far, nothing is known about Σ , not even whether it needs an input itself or not.

In sum, we are left with the observation that the proposed definitions of input do not offer a satisfactory means to define the notion of candidate set, or that they at least face severe conceptual problems. Thus, the first legitimation for inputs in syntax is undermined. In the next section, we turn to the second legitimation for inputs in syntax: the existence of faithfulness constraints.

4. Inputs in Syntax: Faithfulness Constraints

Faithfulness constraints prohibit a deviation of output forms from the input; hence, they crucially rely on the notion of input. However, closer inspection reveals that syntactic faithfulness constraints as they have been proposed in recent work can be reanalyzed as constraints on outputs throughout. This is so because syntax is information preserving in the sense that pieces of information that are attributed to the input (and then referred to by faithfulness constraints) are still accessible in the output and can therefore be referred to by output constraints. Hence, they do not constitute an argument for the existence of inputs.¹⁸

4.1. Dependency Constraints

Constraints of the DEP-family penalize insertion of material into the output that is not present in the input. The classical case in phonology is epenthesis. For

¹⁷This conceptual argument is only valid as long as we assume that input and output are not the same type of object, e.g., partially/highly structured input vs. fully structured output. An alternative view is to assume that GEN relates a fully specified syntactic structure S_i (an input which is an element of the universe of syntactic structures) to a (possibly non-finite) set of fully specified syntactic structures $\{S_1, S_2, \dots, S_n\}$ (a set of outputs). This relation defines the candidate set for every S_i . In this case, input and output are of the same type (namely, fully specified structures). The universe of syntactic structures is in turn generated by an input-free system, for instance a context-free grammar. Such a system (see Vogel 2001a, Wilson 2001) does without the additional generator Σ and is therefore not susceptible to the objection just made. However, given that input and output are not in any way derivationally related in such an approach, the distinction between input and output in the classical sense becomes blurred, and it is not quite clear anymore to what extent the notion of input continues to play a crucial role in the theory. As a matter of fact, it seems to us that such an approach may ultimately be construed as an instantiation of an input-free system itself, with reference to inputs replaced by reference to (possibly virtual) outputs that define candidate sets.

¹⁸Our main concern here is to reconstruct different theories without making use of the notion of faithfulness. In order to compare one version of a theory involving faithfulness with another version of that theory without faithfulness, it is useful to keep all other details of the theory constant and vary only this one aspect. Accordingly, we will not try to improve or comment on the linguistic plausibility of the analyses that we discuss, even if this means that the outcome of our reconstruction may look somewhat unusual in some cases.

syntax, we consider two examples. The constraint FULL INTERPRETATION from Grimshaw (1997a: 374/383) is a first case in point. Grimshaw introduces FULL-INT to make the insertion of auxiliary *do* a costly operation.¹⁹ The nature of this constraint can be understood in different ways:

- (30) FULL-INT
- a. Insertion of expletives is not allowed.
 - b. Expletives are not allowed.
 - c. Lexical conceptual structure is parsed.

If expletives do not appear in the input, FULL-INT is a faithfulness constraint according to (30-a). However, (30-b) derives the same effects and is not a faithfulness constraint. Grimshaw assumes that *do* does not appear in the input, and she presupposes (30-c), which makes reference to properties of *do*. So, (30-c) cannot refer to elements in the input, but it refers to elements that are taken from the lexicon – input-externally – and are inserted as expletives. Hence, under this interpretation FULL-INT is not a faithfulness constraint either.

The second constraint we want to examine is STAY (Grimshaw 1997a: 374; Legendre, Smolensky and Wilson 1998: 11), which penalizes movement.

- (31) STAY
Trace is not allowed.

STAY can be understood as a DEP-constraint if one assumes that traces are not part of the input. However, it is just as well possible to think of it as a constraint on outputs. Given that these results can be viewed as representative, we can conclude that DEP-constraints are superfluous in syntax.

4.2. Maximality Constraints

Constraints of the MAX-family penalize the deletion of input material in the output. The first case we want to take a look at is RECOVERABILITY from Pesetsky (1998: 342):

- (32) RECOVERABILITY
A syntactic unit with semantic content must be pronounced unless it has a sufficiently local antecedent.

At first sight, this looks like a genuine faithfulness constraint. However, there is some doubt as to whether we are dealing with a plausible candidate for faithful-

¹⁹Also see Müller (1997) for the same reasoning with respect to the *wh*-expletive *was* ('what') in German scope marking constructions.

ness in syntax: First, RECOVERABILITY is never violated by any optimal candidate in Pesetsky’s analysis.²⁰ Second, Chomsky (1965: 255) convincingly argues that RECOVERABILITY should be viewed as a precondition for syntax in general, not as a constraint of the grammar. Third, in Pesetsky’s analysis this constraint is not concerned with syntax proper but with the syntax-PF interface, a domain that is not necessarily information preserving.

The second case of a possible MAX-constraint concerns PARSE from Grimshaw and Samek-Lodovici (1998), which is violated by Pro-drop. The definition is given in (33-a).

- (33) PARSE
- a. Pronouns in the input must be realized in the output.
 - b. *pro* is not allowed.
 - c. All arguments of the verb are realized in the syntax.

reanalysis of this is that Pro-drop does not involve complete deletion of the argument in question but the occurrence of an empty pronominal *pro*; cf. (33-b). This version of PARSE again is a pure constraint on outputs without any reference to the input. Yet another possible reformulation, which is arguably closer to the spirit of the original analysis of Grimshaw and Samek-Lodovici and which nevertheless does not make use of faithfulness, is (33-c). The only assumption that this reconstruction of (33-a) requires is that the subcategorization frame (or θ -grid) of a verb is accessible in syntax,²¹ and this assumption strikes us as fairly uncontroversial.²²

4.3. Identity Constraints

The family of IDENT (“identity”) constraints penalizes the change of features on the way from input to output. A member of this constraint family in syntax is the projection principle (see Chomsky 1981). (34-a) is an input-based formulation of the PROJECTION PRINCIPLE as it can be found in the optimality theoretic analyses in Schmid (1999) and Müller (2000a chapter 2).

²⁰In his footnote 31, Pesetsky (1998) mentions one case that may suggest that RECOVERABILITY is violable, but the case is never discussed explicitly.

²¹An anonymous TLR-reviewer suggests that the step from (33-a) to (33-c) just translates syntax/syntax faithfulness into lexicon/syntax faithfulness and therefore does not gain anything. However, if lexical information is indispensable in the syntax anyway, as is assumed in nearly all syntactic theories we are aware of (but see the next footnote), i.e., if lexical information is also present if the constraint in question is (33-a), the above step *is* a step towards a more parsimonious theory.

²²The case may be different in radically minimalist checking approaches, though. In such an approach, merging of an argument with the predicate “checks off” a subcategorization feature of the predicate. When all arguments have been merged in the syntax, all subcategorization features are checked off and are therefore no longer visible.

- (34) PROJECTION PRINCIPLE
- a. Subcategorization features in the input must be satisfied in the output.
 - b. Subcategorization features of lexical items must be satisfied in the syntax.

This constraint is violated non-fatally in the so-called IPP/“Ersatzinfinitiv” construction in German (cf. (35)), and in the case of German R-pronouns (cf. (36)).

- (35) a. ... dass er das *hat*_{:[Part]} lesen wollen
 that he this has to read to want
 ‘that he wanted to read this’
- b. *... dass er das lesen *gewollt hat*_{:[Part]}
 that he this to read wanted has

In (35-a,b) the auxiliary *hat* subcategorizes for the participle *gewollt* (this is indicated by the feature “:[Part]”). However, the infinitival form *wollen* appears in the optimal output (35-a). In (36-a), the R-pronoun *da* is realized in the syntax although the preposition *für* subcategorizes for an NP (see (36-a,b)).

- (36) a. ... dass er *dafür*_{:[NP]} gewesen ist
 that he there for been is
 ‘that he voted for this’
- b. *... dass er *für*_{:[NP]} es gewesen ist
 that he for it been is

Since the subcategorization frame of a lexical item is still recoverable from the syntactic occurrence of that item, it is possible to reanalyze PROJECTION PRINCIPLE as the constraint on outputs in (34-b).

As another example, consider faithfulness in subordination, as discussed in Baković and Keer (2001). Baković and Keer postulate that a property like subordination is marked in the input, and that there is a constraint FAITH[COMP] that penalizes deviation in the output from the information marked in the input; see (37-a).

- (37) FAITH[COMP]
- a. A candidate must realize the [\pm Comp]-specification of the input. ([+Comp]/input \rightarrow [+Comp]/output; [-Comp]/input \rightarrow [-Comp]/output)
 - b. The [\pm Comp]-subcategorization of lexical elements must be observed in the syntax.

The optimal candidates in (38) are faithful to the specification of subordination in the input.

- (38) a. I think_[-Comp] [IP John will leave]
 b. I think_[+Comp] [CP that [IP John will leave]]

However, in cases where a higher ranked constraint must be satisfied (PURE-EP, in the case at hand; see Grimshaw 1997a), FAITH[COMP] is no longer decisive, i.e., it can be violated (cf. the well-formed candidate (39-c)).

- (39) a. *I think_[-Comp] [IP most of the time [IP John will leave]]
 b. *I think_[+Comp] [IP most of the time [IP John will leave]]
 c. I think_[-Comp] [CP that [IP most of the time [IP John will leave]]]
 d. I think_[+Comp] [CP that [IP most of the time [IP John will leave]]]

Baković and Keer assume that the [\pm Comp]-specification is marked on the embedded sentence in the input, but one can think of it as being marked on the embedding verb as well (cf. (37-b)).²³ Then FAITH[COMP] need no longer be viewed as a faithfulness constraint.

The analysis of Romance clitics in Grimshaw (1997b: 194) also crucially relies on faithfulness constraints of the MAX-/IDENT-family. Again it can be shown that they can easily be replaced by constraints on outputs. One phenomenon Grimshaw discusses concerns “floating number” in certain clitic sequences in some dialects of American Spanish, where the plural feature of a clitic in the input is expressed on a different clitic in the output. If the reflexive clitic *se*, which is unspecified for number, realizes a plural argument, plurality is expressed by the following clitic irrespective of whether the second clitic itself represents a plural argument or not. In order to account for this choice of clitics, Grimshaw introduces the two constraints MAX-NUM and IDENT-NUM (cf. (40-a) and (41-a)). If MAX-NUM is higher ranked than IDENT-NUM, the number will float; see tableau T₉. In contrast, the ranking IDENT-NUM \gg MAX-NUM makes the correct predictions for languages with no floating number.²⁴

- (40) MAX-NUM
 a. For every number feature in the input there must be an identical number feature in the output.

²³As a matter of fact, this has been suggested in Legendre, Wilson, Smolensky, Homer, and Raymond (1995: 618).

²⁴We assume here that the realization of the abstract feature “sg” as morphologically unmarked violates neither MAX-NUM nor IDENT-NUM. This is different from the assumptions behind tableau (41) in Grimshaw (1997b: 194), but it can be shown that Grimshaw’s analysis also works under our assumptions.

- b. Every abstract number feature must be morphologically realized.

(41) IDENT-NUM

- a. Number features in the input must be identical to their correspondent number features in the output and vice versa.
- b. The morphological realization of number of each lexical item must match its abstract number feature.

T₉: Number Floating in American Spanish

Input: Clitic _[pl] . . . Clitic _[sg]	MAX-NUM	IDENT-NUM
O ₁ : se _{unmarked} . . . lo _{unmarked}	*!	*
☞ O ₂ : se _{unmarked} . . . los _{plural}		**

If the two constraints in (40-a) and (41-a) are to be replaced with constraints that do not rely on the input, the output will have to be modified in such a way that it encodes the targeted number features of each clitic, i.e., the number features according to which the clitic has to be interpreted independently of its morphological realization. We might, for instance, imagine abstract number features that are associated with each clitic that realizes a plural argument. Thus, it would be possible to distinguish between morphologically realized and targeted number. The constraints in (40-a) and (41-a) could then be replaced with the constraints in (40-b) and (41-b), and the result would be exactly the same as in Grimshaw’s original analysis.

The next example is taken from Legendre, Smolensky and Wilson (1998) (see also section 3.2.2.). The original constraint PARSE SCOPE (cf. (42-a)), which penalizes deviation from the scope information given in the input, can be reformulated as in (42-b), which relies on empty scope markers in output structures (see Williams 1986):

(42) PARSE SCOPE

- a. Scope that is marked in the input has to be realized by chain formation in the output.
- b. Every abstract scope marker must be part of a (non-trivial) chain at S-structure.

Recall that in this system the input contains a target LF, but unfaithful outputs can deviate from that target resulting in a different LF. If one just canceled the input, the information about the target would not be recoverable from the output. Again, the standard strategy can be used to overcome the problem: The output is enriched.

The target LF in the input is encoded by abstract scope markers (“[*S* -]” in the following tableaux) in the output candidates. One then distinguishes between the target LF and the actual LF (the one which is finally interpreted) of each output. As a result, T_{10} – T_{12} are completely parallel to their input-dependent counterparts T_6 – T_8 , without reference to inputs being necessary.

T_{10} : *Wh-Islands for Wide Adjunct-Scope in Chinese*

	SELECT	BAR ² _[-ref]	PARSE SCOPE
O_1 : [<i>S</i> Q_1] ... $V_{[+wh]}$ [<i>CP</i> t'_1 ... wh_1 ...]		*!	
$\rightarrow O_2$: [<i>S</i> -] ... $V_{[+wh]}$ [<i>CP</i> Q_1 ... wh_1 ...]			*

T_{11} : *Narrow Adjunct-Scope in Chinese*

	SELECT	BAR ² _[-ref]	PARSE SCOPE
$\rightarrow O_1$: ... $V_{[+wh]}$ [<i>CP</i> [<i>S</i> Q_1] ... wh_1 ...]			
O_2 : Q_1 ... $V_{[+wh]}$ [<i>CP</i> [<i>S</i> -] ... wh_1 ...]		*!	*

T_{12} : *Declaratives and Wide Adjunct-Scope in Chinese*

	SELECT	BAR ² _[-ref]	PARSE SCOPE
$\rightarrow O_1$: [<i>S</i> Q_1] ... $V_{[-wh]}$ [<i>CP</i> ... wh_1 ...]		*	
O_2 : [<i>S</i> -] ... $V_{[-wh]}$ [<i>CP</i> Q_1 ... wh_1 ...]	*!		*

We can still assume the same criterion for the definition of candidate set: Two competing candidates must have the same target LF.

The same holds for another member of the IDENT-family: the constraint PARSE WH that is also assumed in Legendre, Smolensky and Wilson (1998). The version that is based on faithfulness (cf. (43-a)) can be reformulated as a constraint on outputs without reference to the input (cf. (43-b)).

- (43) PARSE WH
- a. If there is a *wh*-feature on XP in the input, then it has to be realized by an operator-variable chain in the output. XP must not be interpreted as an indefinite.
 - b. A *wh*-element must be part of an operator-variable chain and must not be interpreted as an indefinite.

4.4. Integrity/Uniformity Constraints

Another family of faithfulness constraints restricts one-to-many relations between input and output segments. The relevant constraint families are called INTEGRITY and UNIFORMITY in McCarthy and Prince (1995a):

- (44) a. INTEGRITY
No element of the input has multiple correspondents in the output.
b. UNIFORMITY
No element of the output has multiple correspondents in the input.

A classical violation of UNIFORMITY in phonology is coalescence. In syntax the analysis of free relative constructions proposed in Vogel (2001a) is a case in point. Vogel assumes that the input is a fully specified syntactic structure. In particular, functional categories and the distribution of functional features are given in the input but can be “rearranged.” The motivation for this assumption is the need to account for the ungrammaticality of some examples with free relative (FR) clauses in German and in general to account for languages that have no FR construction. Thus, while the German free relative clause in (45-a) can freely be paraphrased with a headed relative construction (HR) (45-b), only the latter is possible in (46):

- (45) a. Ich folge wem ich vertraue
I follow who_{dat} I trust
'I follow whoever I trust'
b. Ich folge einem dem ich vertraue
I follow one_{dat} who_{dat} I trust
'I follow someone I trust'
- (46) a. *Ich folge wen ich mag
I follow who_{acc} I like
'I follow whoever I like'
b. Ich folge einem den ich mag
I follow one_{dat} who_{acc} I like
'I follow someone I like'

The basic assumption is that FRs and HRs are in competition. Since both structures are possible in German and since it is implausible that they have identical constraint profiles, it is necessary to encode the minimal differences between them in the input and let a faithfulness constraint favor the input-faithful candidate. By assumption, a FR and its corresponding HR are identical in their functional features. The features of the FR pronoun, for instance *wem* in (45-a), are split off and make up their own functional projections in a HR like (45-b). The candidate set is then restricted to clauses that are identical in functional features, but they can vary

as to how these features are distributed. Depending on the relation between input and output, we either have a violation of INTEGRITY (if the feature bundle of a single functional head in the input is split and distributed over several functional heads in the output), or a violation of UNIFORMITY (if the features of two or more functional heads in the input are merged into one functional head in the output).

The two features in question are [REL], which represents the relative operator, and [REF], which denotes the introduction of a new discourse referent. In a FR the two features are bundled together. However, in a HR [REF] is realized by the “head,” and [REL] is realized by the relative pronoun; see (47-ab).

- (47) a. FR pronoun: $[_{D^0} [[REF][REL]]]$
 b. HR: $[_{D^0} [REF]] \dots [_{CP} [_{DP} [_{D^0} [REL]]]] \dots]$

An input-free system can mimic the processes of splitting and merging of feature bundles by assuming feature movement. When the feature bundle of a FR pronoun is split to yield a HR structure, the moved feature leaves behind a trace:

- (48) a. FR as input: $[_{D^0} [[REF][REL]]]$
 b. Unfaithful candidate: HR
 $[_{D^0} [REF]] \dots [_{CP} [_{DP} [_{D^0} [[t_{REF}][REL]]]]] \dots]$

These are the candidates for a FR competition. Those for a HR competition start with a HR structure and merge the features in the unfaithful FR candidate:

- (49) a. HR as input: $[_{D^0} [REF]] \dots [_{CP} [_{DP} [_{D^0} [REL]]]] \dots]$
 b. Unfaithful candidate: FR
 $[_{D^0} [t_{REF}]] \dots [_{CP} [_{DP} [_{D^0} [[REF][REL]]]]]$

We can then assume a constraint that bans feature movement (see Roberts 1998):

- (50) STAY-F
 Feature movement is not allowed.

the same work as an input-output faithfulness constraint on the distribution of functional features. However, it has to be ensured that (49-b) and (48-a) (and likewise (48-b) and (49-a)) do not take part in the same competition. This has to be achieved by the criterion that defines the candidate set.²⁵

²⁵Again, we stress that an appropriate definition of candidate set is far from trivial. Just note in passing that Vogel’s analysis seems to require a rather narrow interpretation of this definition in terms of same LFs (see footnote 16): “same LFs” cannot simply mean “identical interpretation” but has to imply “identical structure”, since, for instance, both (49-b) and (48-a) have the same semantics but one contains an additional trace in contrast to the other.

To sum up this section, we have shown that syntactic faithfulness constraints can systematically be reanalyzed as constraints on outputs that do not refer to the notion of input. Even though the individual analyses that we considered differ both with respect to their theoretical orientation and the empirical domain that they cover, there is a single unifying reason why an input-free reformulation of the pertinent constraints is possible throughout: Syntactic output representations are richly structured objects that can provide all the information that faithfulness constraints locate in the input. More generally, we have shown in the last two sections that the input is not needed for the definition of candidate sets in syntax (which can and, as we have argued, should be defined by an independent criterion: see footnote 16), and that it is not needed for faithfulness constraints in syntax (because the latter can all be reformulated as constraints on outputs). Candidate sets and faithfulness provided the two main empirical legitimations for inputs in syntax. Other empirical motivations are not in sight.²⁶ Hence, we may conclude that inputs can (and, for reasons of conceptual parsimony, arguably should) be dispensed with in optimality theoretic syntax.²⁷ However, recall from section 2. that this result does not hold for phonology. In the final section, we derive this asymmetry between phonology and syntax from independently motivated properties of the two systems.²⁸

5. Why Phonology and Syntax are Different

We would like to contend that there is a crucial difference between phonology and syntax that forces the existence of an input in the former type of system, but not in the latter: Syntax is information preserving, phonology is not. To make this claim clearer: Syntactic systems can be easily construed in a way such that no information is lost during the derivation. All properties of the input are recoverable from the output. However, in phonology this is not always the case.

Consider the prototypical syntactic transformation: movement. If a head's argument is moved away from the head in the syntax, the output-tree still contains the information that the argument was there before movement applied: It is encoded in the selectional properties of the head (usually, and in many cases redundantly, also in traces). That is, the atoms of syntax, i.e., words/morphemes,

²⁶Note, e.g., that an input-free syntax implies that syntactic input optimization cannot exist. Indeed, to the best of our knowledge, no convincing argument has yet been made for the necessity of input optimization in syntax; see Müller (2000a) for overview and discussion.

²⁷Note that something parallel has been proposed in the minimalist program for the concept of numeration; see, e.g., Frampton and Gutman (1999).

²⁸See Bromberger and Halle (1989) for a related discussion; however, although their conclusion is similar to ours, their arguments and assumptions are to some extent incompatible with the view put forth below.

have relational properties. One can only learn about the properties of a head (e.g., [+transitive]) by looking at its behavior within a syntactic context. In contrast, there are no selectional relations between the segments of a phonological representation. Properties of phonemes are never relational, i.e., one can read off all the features of a phoneme (e.g., [+labial]) by merely looking at the phoneme itself. The relational behavior of phonemes is determined exclusively by the phonological constraints/rules. As a consequence, if a transformation has applied in phonology, e.g., deletion/epenthesis, this is not necessarily recoverable from the output representation.

The above observation is related to another property of syntax: its compositional interpretation. A head and its arguments are standardly interpreted by the compositional rule of functional application. On the other hand, the meaning of (non-complex) words is not compositional. As a consequence, whereas for syntax it is in principle possible to define the notion of candidate set by an input-independent criterion, this seems rather hard for phonology. Let us assume that two phonological forms F_1 and F_2 should only compete if they encode the same meaning. Now, consider the output /slɔt/ (*slot*). A potential competitor would be /lɔt/ (*lot*), where the initial /s/ has been deleted. This competitor would lose against /slɔt/ due to a fatal violation of faithfulness.²⁹ But /lɔt/ is an optimal output all by itself. It therefore must be optimal in another competition. One would like to say that /lɔt/ is the winner of a competition where the candidates are exponents of the meaning [lot], but it is a loser in a competition where they are exponents of [slot]. Given this reasoning, however, it is not clear how to avoid competition of synonyms like /θɹɛ:t/ (*threat*) and /mɛnəs/ (*menace*). Since they express the same meaning, they should compete.³⁰ However, they are both optimal outputs, despite exhibiting different constraint profiles.

The problem is that on the word/morpheme level, the association of form and meaning is fixed, and it is not transparent in a compositional way whether the variation of one phonological feature triggers the shift from one lexical item to another, i.e., whether the meaning changes. Hence, a criterion for the candidate set that refers exclusively to lexical meaning is not flexible enough. However, a criterion that makes use of both form and meaning is nothing but an UR input. As a consequence, the input in phonology is necessary not only for reasons of faithfulness (see section 2.3.1.), but also for the definition of candidate set.

Furthermore, the fact that there are opacity effects in phonology can be taken as direct evidence for our hypothesis.³¹ In cases that involve opacity, it is impos-

²⁹Assume for the sake of argument that it is possible to reconstruct the relevant faithfulness constraint without recourse to the input.

³⁰One could claim that synonyms never mean the same thing. This might be true for some synonyms, but we doubt that it is generally the case.

³¹Actually, it has been proposed that opacity effects should be treated in terms of more abstract

sible to infer the underlying representation from the corresponding grammatical output and the given constraints. This is a typical case where information has been lost on the way from the UR to the output. Interestingly, we do not know of any opacity effects in syntax. All this amounts to the same conclusion: Inputs can be dispensed with in syntax but not in phonology because syntax is information preserving and phonology is not.³²

Universität Stuttgart (FH, TS)
Universität Tübingen (SF)
IDS Mannheim (GM)
Universität Potsdam (RV)
Aarhus Universitet (SV)

References

- Ackema, Peter, and Ad Neeleman (1998). Optimal Questions. *Natural Language and Linguistic Theory* 16: 443-490.
- Archangeli, Diana and D. Terence Langendoen (eds.) (1997). *Optimality Theory: An Overview*. London: Blackwell.

representations, see Goldrick (2000) and Bye (2001). Since abstract representations tend to make the input redundant, this may be seen as an argument against input in phonology, contrary to the claim made here. However, since there are other techniques available that deal with opacity in optimality theory (see Kager 1999, McCarthy 1999b, Wilson 2000), we take it that our hypothesis is tenable.

³²A remark is in order concerning the scope of this result. Note that this article only dealt with versions of optimality theoretic syntax that descend from the principles and parameters approach. This means that we do not intend to make any argument for systems that are based on another architecture, like lexical functional grammar (LFG). The basic c-structure/f-structure architecture of LFG strongly suggests the existence of an input in an optimality theoretic version of LFG (see Bresnan 2000, Sells 2002, and Kuhn 2001, among others). Furthermore, problems that arise in a principles and parameters-based optimality theoretic syntax may not find their counterpart in an LFG-based system and vice versa. The two lines of syntactic research are generally quite different in both the “pure” and the optimality theoretic versions. Although a comparison of the two systems against the background of the input discussion might yield interesting results, we cannot go into this matter here, for reasons of space and coherence. The same caveat holds for certain other varieties of optimality theoretic syntax like, for instance, bidirectional optimization (cf. Blutner 2000, Kuhn 2001, Lee 2001, Vogel 2001b).

Furthermore, our arguments only apply to parallel optimization, not to approaches that depend on serial optimization (see Ackema and Neeleman (1998: 478), Müller 2000b, Heck and Müller 2000, Fanselow and Ćavar 2001, and Fischer 2001 for syntactic applications, and McCarthy 1999a, Mester 1999, and Rubach 2000 on the issue in phonology). In serial optimization, the optimal output becomes the input for the next optimization procedure, etc. Such a system cannot be input-free for principled reasons. However, the system provides the input automatically, except for the very first “basic” input, which starts the whole computation. Still, the abolition of this basic input may cause problems (see, for instance, the analysis of successive-cyclic *wh*-movement in Heck and Müller 2000, which makes crucial use of a numeration as basic input).

- Baker, Carl L. (1970). Notes on the Description of English Questions: the Role of an Abstract Question Morpheme, *Foundations of Language* 6: 197–219.
- Baković, Eric and Ed Keer (2001). Optionality and Ineffability. In Géraldine Legendre, Jane Grimshaw and Sten Vikner (eds.), *Optimality-Theoretic Syntax*. Cambridge, Massachusetts: MIT Press, 97–112.
- Benua, Laura (1997). Transderivational Identity: Phonological relations between words. Doctoral dissertation, University of Massachusetts.
- Blutner, Reinhard (2000). Some Aspects of Optimality in Natural Language Interpretation. Ms., Humboldt Universität Berlin.
- Bresnan, Joan (2000). Optimal Syntax. In Joost Dekkers, Frank van der Leeuw and Jeroen van der Weijer (eds.), *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford: Oxford University Press, 334–385.
- Bromberger, Sylvain and Morris Halle (1989). Why Phonology is Different. *Linguistic Inquiry* 20: 51–70.
- Burzio, Luigi (1996). Surface Constraints versus Underlying Representation. In Jacques Durand and Bernard Laks (eds.), *Current Trends in Phonology: Models and Methods*. University of Salford Publications. 97-122
- Bye, Patrik (2001). Virtual Phonology: Rule Sandwiching and Multiple Opacity in North Saami. Doctoral Dissertation, University of Tromsø.
- Chomsky, Noam (1965). *Aspects of the Theory of Syntax*. Cambridge, Massachusetts: MIT Press.
- Chomsky, Noam (1981). *Lectures on Government and Binding*. Dordrecht: Foris.
- Chomsky, Noam (1995). *The Minimalist Program*. Cambridge, Massachusetts: MIT Press.
- Fanselow, Gisbert and Damir Ćavar (2001). Remarks on the Economy of Pronunciation. In Gereon Müller and Wolfgang Sternefeld (eds.), *Competition in Syntax*. Berlin: Mouton de Gruyter, 107–150.
- Fischer, Silke (2001). Reanalyzing Reconstruction Effects: An Optimality-Theoretic Account of the Relation between Pronouns and R-Expressions. To appear in Proceedings of ConSOLE 9.
- Frampton, John, and Sam Gutman (1999). Cyclic Computation. *Syntax* 2: 1-27.
- Goldrick, Matthew (2000). Turbid Output Representations and the Unity of Opacity. In Masako Hirotani and Andries Coetzee and Nancy Hall and Ji-Yung Kim (eds.), *Proceedings of NELS 30*. Amherst, Massachusetts: GLSA, 231–245.
- Grimshaw, Jane (1997a). Projection, Heads, and Optimality. *Linguistic Inquiry* 28: 373–422.
- Grimshaw, Jane (1997b). The Best Clitic: Constraint Conflict in Morphosyntax. In Liliane Haegeman (ed.), *Elements of Grammar*. Dordrecht: Kluwer, 169–196.
- Grimshaw, Jane and Vieri Samek-Lodovici (1998). Optimal Subjects and Subject Universals. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis

- and David Pesetsky (eds.), *Is the best good enough? Optimality and competition in syntax*. Cambridge, Massachusetts: MIT Press, 193–219.
- Hayes, Bruce (1999). Phonological Restructuring in Yidiñ and its theoretical consequences. In Ben Hermans and Marc van Oostendorp (eds.), *The Derivational Residue in Phonological Optimality Theory*. Amsterdam: John Benjamins, 175–206
- Heck, Fabian and Gereon Müller (2000). Repair-Driven Movement and the Local Optimization of Derivations. Ms., Universität Stuttgart and IDS Mannheim.
- Kager, René (1999). *Optimality Theory*. Cambridge: Cambridge University Press.
- Kenstowicz, Michael and Charles Kisseberth (1979). *Generative Phonology*. New York: Academic Press.
- Kuhn, Jonas (2001). Formal and Computational Aspects of Optimality-Theoretic Syntax. Doctoral dissertation, Universität Stuttgart.
- Lee, Hanjung (2001). Optimization in Argument Expression and Interpretation: A Unified Approach. Doctoral dissertation, Stanford University.
- Legendre, Géraldine, Colin Wilson, Paul Smolensky, Kristin Homer and William Raymond (1995). Optimality and Wh-Extraction. In Jill Beckman, Laura Walsh-Dickie and Suzanne Urbanczyk (eds.), *Papers in Optimality Theory*. Amherst, Massachusetts:GSLA, 607–634.
- Legendre, Géraldine, Paul Smolensky and Colin Wilson (1998). When is less more? Faithfulness and Minimal Links in WH-Chains. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky (eds.), *Is the best good enough? Optimality and competition in syntax*. Cambridge, Massachusetts: MIT Press, 249–289.
- McCarthy, John (2000a). Harmonic serialism and harmonic parallelism. In Masako Hirotani and Andries Coetzee and Nancy Hall and Ji-Yung Kim (eds.) *Proceedings of NELS 30*. Amherst, Massachusetts: GLSA, 501–524.
- McCarthy, John (1999b). Sympathy and Phonological Opacity, *Phonology* 16: 331–399.
- McCarthy, John and Alan Prince (1993). Prosodic Morphology I – Constraint Interaction and Satisfaction. Ms., University of Massachusetts, Amherst and Rutgers University.
- McCarthy, John and Alan Prince (1995a). Faithfulness and Reduplicative Identity. In Jill Beckman, Laura Walsh-Dickie and Suzanne Urbanczyk (eds.), *Papers in Optimality Theory*. Amherst, Massachusetts: GSLA 249–384.
- McCarthy, John and Alan Prince (1995b). Faithfulness and Reduplicative Identity in Prosodic Morphology. Ms., University of Massachusetts, Amherst and Rutgers University.
- Mester, Armin (1999). Weak Parallelism: Serial and Parallel Sources of Opacity in OT. Ms., University of California, Santa Cruz.
- Müller, Gereon (1997). Partial Wh-Movement and Optimality Theory. *The Lin-*

- guistic Review* 14: 249–306.
- Müller, Gereon (2000a). *Elemente der optimalitätstheoretischen Syntax*. Tübingen: Stauffenburg.
- Müller, Gereon (2000b). Shape Conservation and Remnant Movement. In Masako Hirotani and Andries Coetzee and Nancy Hall and Ji-Yung Kim (eds.) *Proceedings of NELS 30*. Amherst, Massachusetts: GLSA, 525–539.
- Pesetsky, David (1998). Optimality Principles of Sentence Pronunciation. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky (eds.), *Is the best good enough? Optimality and competition in syntax*. Cambridge, Massachusetts: MIT Press, 337–383.
- Pinsker, Hans Ernst (1959). *Historische Englische Grammatik*. München: Max Hueber.
- Prince, Alan and Paul Smolensky (1993). Optimality Theory. Constraint Interaction in Generative Grammar. Ms., Rutgers University and University of Colorado at Boulder.
- Roberts, Ian (1998). Have/Be Raising, Move F, and Procrastinate. *Linguistic Inquiry* 29: 113–125.
- Ross, Alan (1965). *Etymology: With Special Reference to English*. London: Methuen.
- Rubach, Jerzy (2000). Glide and Glottal Stop Insertion in Slavic Languages: A DOT Analysis. *Linguistic Inquiry* 31: 271–317.
- Schmid, Tanja (1999). West Germanic Infinitivus Pro Participio (IPP) constructions in Optimality Theory. In Tina Cambier-Langeveld, Anikó Lipták, Michael Redford and Erik Jan van der Torre (eds.), *Proceedings of ConSOLE VII*. Leiden: Sole, 229–244.
- Sells, Peter (2002). Structure, Alignment, and Optimality in Swedish. Stanford: CSLI publications.
- Sternefeld, Wolfgang (1997). Comparing Reference Sets. In Chris Wilder, Hans-Martin Gärtner and Manfred Bierwisch (eds.), *The Role of Economy Principles in Linguistic Theory*. Berlin: Akademie-Verlag, 81–114.
- Tesar, Bruce and Paul Smolensky (2000). *Learnability in Optimality Theory*. Cambridge, Massachusetts: MIT Press.
- Vikner, Sten (1997). The Interpretation of Object Shift, Optimality Theory, and Minimalism. *Working Papers in Scandinavian Syntax* 60: 1–24.
- Vogel, Ralf (2001a). Case Conflict in German Free Relative Constructions. An Optimality Theoretic Treatment. In Gereon Müller and Wolfgang Sternefeld (eds.), *Competition in Syntax*. Berlin: Mouton de Gruyter, 341–376.
- Vogel, Ralf (2001b). Word Order Freezing as MLC Effect in Bidirectional OT. Ms., Universität Stuttgart.
- Williams, Edwin (1986). A Reassignment of the Functions of LF. *Linguistic Inquiry* 17: 265–300.
- Wilson, Colin (2000). Consonant Cluster Neutralization and Targeted Constraints.

Doctoral dissertation, Johns Hopkins University, Baltimore.
Wilson, Colin (2001). Bidirectional Optimization and the Theory of Anaphora.
In Jane Grimshaw, Géraldine Legendre and Sten Vikner (eds.), *Optimality
Theoretic Syntax*. Cambridge, Massachusetts: MIT Press, 465–507.