Constraints in Syntax

[6] Optimality-Theoretic Syntax

SoSe 2005, Universität Leipzig
Gereon Müller
gereon.mueller@uni-leipzig.de

1. Basic Concepts

Note:
By definition, an optimality-theoretic syntax takes the general form in (1), with the grammar divided into a Gen part that creates the competing candidates, and a H-Eval part that selects the optimal candidate(s).

(1) *Structure of a competition-based syntax*:
  a. Gen creates the candidate set $\{C_1, C_2, ...\}$.
  b. H-Eval determines the optimal candidate(s) C_i (C_j, ...) in $\{C_1, C_2, ...\}$.

Recall:
The notion of optimality in a minimalist syntax is a comparatively simple one: Optimality is determined by a small set of simple transderivational or translocal economy constraints, as in (2).

(2) *Optimality in minimalist syntax*:
A derivation D_i is optimal iff there is no derivation D_j in the same reference set that is preferred over D_i by a transderivational or translocal constraint.

But:
In optimality-theoretic syntax, there is only one translocal or transderivational constraint that determines optimality: Optimal (and grammatical) is a candidate that has the best constraint profile – or, more precisely, a candidate for which there is no competitor that has a better constraint profile; cf. (3).
(Whether this optimality principle is transderivational or translocal depends on whether the competing candidates are assumed to be derivations or output representations.)

(3) *Optimality in optimality-theoretic syntax*:
A candidate C_i is optimal (= grammatical) iff there is no candidate C_j in the same candidate set that has a better constraint profile.

(4) *Constraint Profile*:
A candidate C_j has a better constraint profile than a candidate C_i iff there is a constraint Con_k such that (a) and (b) hold:
  a. C_j satisfies Con_k better than C_i.

b. There is no constraint Con_l ranked higher than Con_k on which C_i and C_j differ.

Note:
C_j satisfies a constraint Con better than C_i iff C_j has fewer violations of Con. This implies the case that C_i violates Con once (or more often), and C_j does not violate Con at all.

Note:
This presupposes that in addition to the local (or global) constraints employed by the Gen component, which are inviolable and unranked, the H-Eval component relies on a system of local (or global) constraints that are violable and ranked (and, by assumption, universal) in order to determine the best constraint profile, hence, optimality. The ranking among the violable local constraints of the H-Eval component is indicated by the symbol $\gg$; the H-Eval constraints themselves are typically written with small capitals. Optimality-theoretic competitions are often illustrated by tables (so-called *tableaux*); optimality of a candidate is indicated by the *pointing finger*: ☞; violation of a local constraint is shown by a star * in the appropriate column of the table; if this violation is fatal for a candidate (i.e., responsible for its suboptimality), an exclamation mark ! is added. In the abstract H-Eval competition in table T_1, in which the candidate set consists of C_1–C_5, C_1 emerges as the optimal candidate: It avoids a violation of the high-ranked constraints A and B (unlike C_3–C_5), and it minimizes a violation of the low-ranked constraint C (unlike C_2). Hence, there is no competing candidate with a better constraint profile than C_1.

T_1: Determining optimality

Candidates	A	B	C
☞C_1			*
C_2			**!
C_3		*!	
C_4	*!		
C_5		*!	*

Constraint reranking = parametrization:
By reranking the constraints B and C in T_1, candidate C_3 would emerge as the optimal candidate. Reranking of constraints forms the basis of the concept of parametrization in optimality-theoretic syntax.

Non-cumulativity:
A further characteristic feature of this approach is that it is essentially non-cumulative; i.e., no number of violations of a low-ranked constraint can outweigh a single violation of a higher-ranked constraint. Thus, suppose that there were an additional, lowest-ranked constraint D in T_1 that C_1 violates, say, five times, and that C_2–C_5 do not violate at all. This would not undermine C_1's optimality.

Candidates and violable constraints:
Something needs to be said about the nature of candidates and candidate sets. Optimality-theoretic syntax is strongly influenced by work in optimality-theoretic phonology. Since the latter is characterized by an orientation that is predominantly representational (cf. Prince & Smolensky (1993) and McCarthy & Prince (1995)), it does not come as a surprise that many approaches in optimality-theoretic syntax postulate that the competing candidates created by Gen are output representations. This holds, e.g., for what can arguably be viewed as the three most influential analyses in optimality-theoretic syntax so far (outside the LFG work by Bresnan, Choi, Sells, and others), viz., Grimshaw (1997), Pesetsky (1998), and Legendre, Smolensky & Wilson (1998). However, there is no inherent reason why the candidates that are subject to optimization should not be syntactic objects of a more complex type, like <D-structure,S-structure,LF> tuples as in the Principles-and-Parameters approach, or, indeed, complete derivations, as in the minimalist program. The choice of candidate type goes hand in hand with the choice of local (or global) constraint type that shows up in the H-Eval part as a violable and ranked: If candidates are representations, constraints will be representational, if candidates are derivations, constraints will be derivational (or global), and if candidates are <D-structure,S-structure,LF> tuples, constraints can take any of the forms sketched in [2].

Candidate sets:
Similarly, candidate sets can be defined in various ways, which of course significantly influences the nature of the competition. Basically, all of the definitions of reference sets in minimalist syntax that have been proposed (see [5], and Sternefeld (1997) for an overview) are also potential definitions of candidate sets in optimality-theoretic syntax. A further influential definition of candidate sets comes from Grimshaw (1997). She postulates that two candidates (S-structure representations) compete iff they are realizations of the same predicate/argument structure and have non-distinct logical forms (or non-distinct interpretation).

Note:
By making optimality depend on an intricate system of violable and ranked constraints, H-Eval – and hence, the concept of competition – becomes even more important than in minimalist syntax and blocking syntax. As a matter of fact, much work in optimality-theoretic syntax tries to minimize the role of the Gen component, and maximize the role of the H-Eval component (but see Pesetsky (1997; 1998) for some cautionary remarks).

What is optimality theorety good for in syntax?
An optimality-theoretic approach gains immediate support in all those contexts where postulating a competition of syntactic objects is initially plausible. This includes, but is by no means confined to, contexts where notions of economy seem to play a role. A prototypical case is one in which the wellformedness of a sentence S_i that exhibits an otherwise

peculiar property seems to depend on the unavailability of another sentence S_j that exhibits the property one would normally expect. Here, S_i is often referred to as a "repair" or "last resort" form; a typical instance is the English *do*-support construction. Accordingly, *do*-support was among the first phenomena to be tackled in optimality-theoretic syntax (see Speas (1995) and Grimshaw (1997)). Most of the constructions discussed in [5] can also be viewed as suggesting an underlying competition; and indeed, they can fruitfully be addressed in optimality-theoretic syntax.

2. Anaphors vs. Pronouns

(5) *Consequences of Principle A*:

a. $[_{CP}$ C $[_{TP}$ $John_1$ $[_{T'}$ Ø $[_{VP}$ t_1 likes $himself_1$]]]]

b. *$[_{CP}$ C $[_{TP}$ $John_1$ $[_{T'}$ Ø $[_{VP}$ t_1 thinks $[_{CP}$ that $[_{TP}$ $Mary_2$ $[_{T'}$ Ø $[_{VP}$ t_2 likes $himself_1$]]]]]]]]

(6) *Consequences of Principle B*:

a. *$[_{CP}$ C $[_{TP}$ $John_1$ $[_{T'}$ Ø $[_{VP}$ t_1 likes him_1]]]]

b. $[_{CP}$ C $[_{TP}$ $John_1$ $[_{T'}$ Ø $[_{VP}$ t_1 thinks $[_{CP}$ that $[_{TP}$ $Mary_2$ $[_{T'}$ Ø $[_{VP}$ t_2 likes him_1]]]]]]]]

Generalization:
By and large, pronouns are allowed to express binding relations in English in just those cases in which anaphors are not allowed to do so. This has been taken to indicate that there is a competition of the two strategies: We do not have to stipulate both Principle A and Principle B. Rather, one of the two constraints (usually Principle A) is adopted, and the other is derived as an elsewhere case by invoking syntactic competition (see Fanselow (1989; 1991), Burzio (1991; 1998), Reinhart (1991), and Richards (1997), among others). The following analysis essentially goes back to Wilson (2001).

(7) a. LOC-ANT^r ("Local Antecedent"):
If a binding domain contains an anaphor, then it must also contain the anaphor's antecedent.

b. REF-ECON^r ("Referential Economy"):
An argument must not have lexical ϕ-feature specification.

(8) *Ranking in English*:
LOC-ANT $\gg$ REF-ECON.

Note:
(i) LOC-ANT is a version of Principle A. Other things being equal, this constraint favours pronouns.
(ii) REF-ECON inherently prefers anaphors to pronouns, assuming that anaphors do not have a lexical ϕ-feature specification, whereas pronouns do.

T_2: Reflexivization

Candidates	LOC-ANT	REF-ECON
☞C_1: $John_1$ likes $himself_1$		
C_2: $John_1$ likes him_1		*!

T_3: Pronominalization

Candidates	LOC-ANT	REF-ECON
C_1: $John_1$ thinks that Mary likes $himself_1$	*!	
☞C_2: $John_1$ thinks that Mary likes him_1		*

3. Complementizer-Trace Effects

Note:
Recall from [3] that the Principles-and-Parameters approach accounts for complementizer-trace effects (= that-trace effects) on a purely local basis, without postulating a competition with the complementizer-less variant from which only the latter would emerge as optimal.

(9) *The ECP account of that-trace effects*:

a. *Who_1 do you think [$_{CP}$ t'_1([+γ]) that [$_{TP}$ t_1([–γ]) will leave]] ?

b. Who_1 do you think [$_{CP}$ t'_1([+γ]) Ø [$_{TP}$ t_1([+γ]) will leave]] ?

Background assumptions:
This view is abandoned in Déprez (1991), which is the basis of the optimality-theoretic account advanced in Grimshaw (1997). As background, Grimshaw assumes that the size of clauses is variable. Clauses are extended projections of V; they are minimally VPs, but they can be TPs, CPs, or functional projections of an even bigger size, depending on the outcome of optimization. Bridge verbs in English permit both CP-embedding (with a complementizer – a declarative CP without a complementizer will typically fatally violate a high-ranked constraint that precludes empty head positions) and TP- or VP-embedding (without a complementizer). In the latter case, TP must be chosen if an auxiliary or *do* is present (i.e., if the need arises to accomodate an additional lexical head); VP otherwise. A co-occurrence of CP- and TP- (VP-) embedding is possible if the two candidates have an identical constraint profile. This implies that the presence of *that* does not violate any constraint, an assumption that Grimshaw (1997) makes even though it is not completely unproblematic (as she acknowledges). The main constraints that are needed in the account of complementizer-trace effects are listed in (10).

(10) a. OP-SPEC^r ("Operator in Specifier"):
Wh-operators must occupy a specifier position from which they c-command all elements of the extended V projection over which they take scope.

b. T-LEX-GOV^r ("Lexical Government of Traces"):
A trace is lexically governed.

c. STAY^r ("Economy of Movement"):
Trace is not allowed.

(11) *Ranking in English*:
OP-SPEC ≫ T-LEX-GOV ≫ STAY.

Note:
STAY is a local version of the translocal economy constraint Fewest Steps; OP-SPEC is a version of the *Wh*-Criterion that has often been postulated in the Principles-and-Parameters approach (see, e.g., Lasnik & Saito (1992)). (However, we have seen that the effects of such a constraint can be derived from postulating [*Q*] features on $C_{[+wh]}$ items.)

Assumption:
Candidates with and without *that* compete.

T_4: Subject wh-movement

Candidates	OP-SPEC	T-LEX-GOV	STAY
C_1: ... who_1 you think [$_{CP}$ that [$_{TP}$ t_1 will leave]]			*
☞C_2: ... who_1 you think [$_{TP}$ t_1 will leave]		*!	*
C_3: ... you think [$_{CP}$ that [$_{TP}$ who_1 will leave]]	*!		
C_4: ... you think [$_{TP}$ who_1 will leave]	*!		

Analysis:
C_3 and C_4 fatally violate OP-SPEC. Both C_1 and C_2 violate STAY, but C_2 violates T-LEX-GOV in addition: t_1 in C_2 is not lexically governed (*that* being unable to do so), whereas t_1 in C_1 is lexically governed (by the matrix V). In contrast, an embedded V governs object traces throughout, irrespective of the presence or absence of a complementizer *that*; hence, T-LEX-GOV is satisfied equally well by C_1 and C_2 in table T_5. Given that C_1 and C_2 do not differ with respect to any other constraint either, optionality of a complementizer is correctly predicted in cases of object extraction, due to an identical constraint profile.

T_5: Object wh-movement

Candidates	OP-SPEC	T-LEX-GOV	STAY
☞C_1: ... who_1 you think [$_{CP}$ that [$_{TP}$ she will invite t_1]]			*
☞C_2: ... who_1 you think [$_{TP}$ she will invite t_1]			*
C_3: ... you think [$_{CP}$ that [$_{TP}$ she will invite who_1]]	*!		
C_4: ... you think [$_{TP}$ she will invite who_1]	*!		

Observation:
Thus far, there is no evidence for treating T-LEX-GOV as a violable constraint in the H-Eval part of the grammar (rather than as an inviolable constraint in the Gen part). Such evidence

can be gained by considering adjunct extraction. In this case, T-LEX-GOV is violated by both candidates involving *wh*-movment (C_1, C_2). However, given that there is no competing candidate that can avoid a violation of T-LEX-GOV without violating a higher-ranked constraint (e.g., a candidate that employs a resumptive pronoun; see below), C_1 and C_2 can emerge as optimal despite this violation.

T_6: Adjunct wh-movement

Candidates	OP-SPEC	T-LEX-GOV	STAY
☞C_1: ... why$_1$ you think [$_{CP}$ that [$_{TP}$ she has left t_1]]		*	*
☞C_2: ... why$_1$ you think [$_{TP}$ she has left t_1]		*	*
C_3: ... you think [$_{CP}$ that [$_{TP}$ she has left why$_1$]]	*!		
C_4: ... you think [$_{TP}$ she has left why$_1$]	*!		

4. Subjacency and Resumptive Pronouns

(12) *Resumptive pronouns in Complex NP Constraint contexts*:

a. the man [$_{CP}$ who(m)$_1$ I saw t_1]

b. *the man [$_{CP}$ who(m)$_1$ I don't believe [$_{DP}$ the claim [$_{CP}$ t'_1 that anyone saw t_1]]]

c. *the man [$_{CP}$ who(m)$_1$ I saw him$_1$]

d. ?the man [$_{CP}$ who(m)$_1$ I don't believe [$_{DP}$ the claim [$_{CP}$ t'_1 that anyone saw him$_1$]]]

Note:
Resumptive pronouns often seem to be possible only as last resort strategies, in cases where traces are blocked. Competition-free approaches to syntax have no obvious means to relate one construction to the other; but the case is different in optimality-theoretic syntax. An optimality-theoretic account of resumptive pronoun strategies is developed in Legendre, Wilson & Smolensky (1998) (on the basis of evidence from Chinese) and Pesetsky (1998) (on the basis of English data comparable to those in (12), as well as evidence from Hebrew, Russian, and Polish). (A similar last resort analysis is given in Hornstein (2000) within the minimalist program.) The details of the analyses differ a lot, but the gist of the explanation is identical; it centers around two constraints like those in (13).

(13) a. CNPC^r ("Complex NP Constraint"):
*... α_1 ... [$_{DP}$ ... [$_{CP}$ ... t_1 ...]] ...

b. RES^r ("Resumptive Pronoun Constraint"):
Resumptive pronouns are prohibited.

(14) *Ranking in English*:
CNPC $\gg$ RES.

Note:
The CNPC prohibits traces in certain (non-local) environments; RES disfavours resumptive

pronouns (i.e., pronouns that have c-commanding co-indexed antecedent in an A-bar position) in general. As with the *wh*-movement construction discussed in the last section, it must be ensured that overt movement of the relative operator takes place in examples like those in (12). Suppose that this is independently taken care of – e.g., by a high-ranked (or inviolable Gen-) constraint demanding deletion of [*rel*].

T_7: Trace vs. resumptive pronoun in transparent contexts

Candidates	CNPC	RES
☞C_1: the man [$_{CP}$ who(m)$_1$ I saw t_1]		
C_2: the man [$_{CP}$ who(m)$_1$ I saw him$_1$]		*!

Analysis:
Both candidates respect CNPC. Consequently, the RES violation incurred by the resumptive pronoun in C_2 becomes fatal, and C_1 is optimal. However, in the competition illustrated in table T_8, C_1 violates the CNPC. In this case, C_2's RES violation is tolerable, and the resumptive pronoun strategy emerges as optimal.

T_8: Trace vs. resumptive pronoun in CNPC contexts

Candidates	CNPC	RES
C_1: the man [$_{CP}$ who(m)$_1$ [$_{TP}$ I don't believe [$_{DP}$ the claim [$_{CP}$ t'_1 that anyone saw t_1]]]]	*!	
☞C_2: the man [$_{CP}$ who(m)$_1$ [$_{TP}$ I don't believe [$_{DP}$ the claim [$_{CP}$ that anyone saw him$_1$]]]]		*

5. Avoid Pronoun

Recall:
As noted in [3], PRO may block overt pronouns in English gerunds. PRO and a lexical pronoun can both occur in principle; however, PRO must be used instead of a lexical pronoun if it can fulfill the Constraint on Control. This was derived by adopting a transderivational/translocal Avoid Pronoun constraint in Chomsky (1981).

(15) *PRO in English gerunds*:

a. John$_1$ would much prefer [PRO$_1$ going to the movie]

b. *John$_1$ would much prefer [PRO$_{2/arb}$ going to the movie]

(16) *Pronouns in English gerunds*:

a. *John$_1$ would much prefer [his$_1$ going to the movie]

b. John$_1$ would much prefer [his$_2$ going to the movie]

(17) CONTROL^r ("Constraint on Control", Manzini (1983)):
If PRO is minimally dominated by a declarative clausal complement α, then it must be

controlled by an antecedent within the minimal CP that dominates α.

(18) *Avoid Pronoun*$^{td/tl}$:
If two derivations D_1 and D_2 are in the same reference set and D_1 uses a lexical pronoun where D_2 uses an empty pronoun, then D_1 is to be preferred over D_2.

Note:
Chomsky's (1981) approach can straightforwardly be translated into optimality theory:
(i) The Constraint on Control in (17) can directly be viewed as an optimality-theoretic constraint.
(ii) The Avoid Pronoun constraint in (18) can be simplified by turning this transderivational/translocal constraint into a local (though violable) one; cf. (19).

(19) *PRON^r ("Avoid Pronoun"):
Pronouns are prohibited.

(20) *Ranking in English*:
CONTROL $\gg$ *PRON.

Candidate sets:
Suppose that candidate sets are defined in such a way that candidates with PRO and candidates with a lexical pronoun can compete, but, crucially, that sentences with different indexings (hence, different logical forms) do not compete. Then, the facts fall into place. The blocking of a lexical pronoun by PRO in cases where CONTROL can be satisfied is illustrated in table T_9.

T_9: PRO vs. pronoun under co-indexing

Candidates	CONTROL	*PRON
C_1: $John_1$ would much prefer [his_1 going to the movie]		*!
☞C_2: $John_1$ would much prefer [PRO_1 going to the movie]		

But:
Table T_{10} illustrates the case where PRO is not co-indexed with the matrix antecedent, thereby violating CONTROL. Here, the *PRON violation incurred by all pronouns is non-fatal, and the pronoun strategy is optimal.

T_{10}: PRO vs. pronoun under contra-indexing

Candidates	CONTROL	*PRON
☞C_1: $John_1$ would much prefer [his_2 going to the movie]		*
C_2: $John_1$ would much prefer [PRO_2 going to the movie]	*!	

6. Open Issues

Note:
Optimality-theoretic syntax inherits the *complexity* problem from minimalist syntax. Candidate sets are typically large (as in Pesetsky (1998)), often infinite (as in Grimshaw (1997)). In addition, there are several open issues that are specific to the optimality-theoretic approach. The focus is on two of these in what follows: inputs and absolute ungrammaticality.

6.1. Inputs and Faithfulness

The input and its functions:
An important optimality-theoretic concept that has played no role so far is the notion of *input*. In optimality theory (cf. Prince & Smolensky (1993)), Gen does not create competing candidates (the *outputs*) freely; rather, it does so on the basis of a given input. In phonology, inputs are underlying representations stored in the lexicon; here, inputs qualify as roughly the same types of objects as outputs. In syntax, it is much less clear what the input might look like (see Archangeli & Langendoen (1996)). The null hypothesis – that the input is a completely articulated potential sentence of the same type as the output candidates – is not unproblematic because it would seem to imply the assumption that all possible sentences are "stored," which cannot possibly be true. To find out what the input in syntax is, it is instructive to consider its theory-internal functions. By and large, there are two. First, the input is standardly taken to define the competition, i.e., candidate sets. Second, the input serves as a basis for *faithfulness* constraints that demand input/output identity and thereby minimize deviations from the input in the optimal output. Let us consider the second function first. Faithfulness constraints can be viewed as global constraints because they cannot be checked by considering solely an output representation or a derivational step.

Faithfulness constraints:
Faithfulness constraints play an important role in phonology. Constraints of the PARSE (or MAX) family prohibit deletion of input material in the output; constraints of the FILL (or DEP) family prohibit insertion of output material that is not part of the input; constraints of the IDENT family prohibit modifying input material. Faithfulness constraints are also adopted in much recent work in optimality-theoretic syntax. The following two constraints are taken from Legendre, Smolensky & Wilson (1998) and Baković & Keer (2001), respectively.

(21) *Syntactic faithfulness constraints*:

a. PARSE[SCOPE]^g:
Scope assignment in the input must be realized by chain formation in the output.

b. FAITH[COMP]^g:
The output value of [±COMP] is the same as the input value.

Note:
(21-a) implies that the input is a more complex object than just a lexical array or a predicate/argument structure; it must be a highly structured representation that exhibits the relative scope of operators. (21-b) presupposes an abstract feature [±COMP] that for present purposes we can assume to be located on a V that selects a proposition. Let us consider candidates that violate these constraints. Suppose that (22-a) is the input for output candidate (22-b), and (22-c) is the input for output candidate (22-d). Legendre, Smolensky & Wilson (1998) assume that (22-b) violates PARSE[SCOPE] because matrix scope for how_1 in the input (22-a) (indicated by $[+wh]_1$) is reduced to embedded scope in the output (again indicated by $[+wh]_1$). Similarly, Baković & Keer (2001) assume that (22-d) violates FAITH[COMP] because a [–COMP] specification in the input contrasts with a [+COMP] specification (hence, a complementizer) in the output.

(22) *Violations of syntactic faithfulness constraints*:

a. $[+wh]_1$ ... $wonder_{[+wh]}$ [$[+wh]_2$ ... $what_2$ ... how_1 ...] (input)
b. You $wonder_{[+wh]}$ [$_{CP}$ $[+wh]_1$ $[+wh]_2$ how_1 John did $what_2$] (output)
c. ... $V_{[-Comp]}$ [...] (input)
d. I think [$_{CP}$ that [$_{PP}$ on him $]_1$ no coat looks good t_1] (output)

A reinterpretation of faithfulness constraints:
At this point, we need not go into the actual analyses in which these constraints play a role (as it happens, both faithfulness violations turn out to be non-fatal, i.e., (22-bd) are optimal). The crucial question is: Is it really necessary to refer to the concept of input here, or is it possible to read the respective violations off the output forms, without any reference to inputs? At least for the cases at hand, the answer is straightforward: By enriching output representations in ways that have independently been proposed, a reference to inputs becomes unnecessary. (22-ab) is a case where the intended matrix scope is not reached by chain formation in the candidate. Employing abstract scope markers (Σ) in S-structure representations (cf., e.g., Williams (1986)), we can equivalently encode this input information in the output, as in (23-a). As for the case in (22-cd), the only assumption that we have to make (which is completely standard) is that selectional properties of lexical heads are accessible in syntax; cf. (23-b).

(23) *Violations of syntactic markedness constraints*:

a. Σ_1 you $wonder_{[+wh]}$ [$_{CP}$ $[+wh]_1$ $[+wh]_2$ how_1 John did $what_2$] (output)
b. I $think_{[-Comp]}$ [$_{CP}$ that [$_{PP}$ on him $]_1$ no coat looks good t_1] (output)

PARSE[SCOPE] and FAITH[COMP] can now be modified in obvious ways, without reference to inputs, as regular local (so-called *markedness*) constraints.

(24) *Faithfulness constraints as markedness constraints*:

a. PARSE[SCOPE]^r (revised):
Scope markers must be reached by chain formation.

b. FAITH[COMP]^r (revised):
Lexical [±COMP] selection requirements must be respected.

Note:
If this result can be generalized, and all syntactic faithfulness constraints can be reanalyzed in this way, we can conclude that these constraints do not support the concept of input anymore. Why should it be that the notion of input is relevant for phonological faithfulness constraints, but not for their syntactic counterparts? The answer, we believe, follows from what appears to be a fundamental difference between syntax and phonology: Syntax is an *information-preserving system* with richly structured output candidates, whereas phonology is a system that is standardly taken to lose information in the course of a derivation (e.g., from input to output), so that reference to an underlying input is necessary in constraints.

Candidate Sets:
With this in mind, consider the other input function noted above, that of defining candidate sets. Since syntactic output candidates are richly structured, all the relevant information that they must share in order to compete can be read off them, independently of what notion of candidate set is adopted; again, this is in sharp distinction to phonology. Hence, reference to inputs seems unnecessary for this purpose in syntax.

Conclusion:
From all this, it seems that one can conclude that it may eventually be possible to dispense with the notion of input in syntax; but further research is needed in this domain (see Heck et al. (2002)).

6.2. Absolute Ungrammaticality

Observation:
Another important open question in optimality-theoretic syntax is how to account for the phenomenon of *absolute ungrammaticality* or *ineffability*, i.e., cases where there does not seem to be a candidate in a candidate set that is grammatical. As an example, consider the following ungrammatical example involving *wh*-extraction across an adjunct island in German:

(25) *A consequence of the Adjunct Condition*:

*Was$_1$ ist Fritz eingeschlafen [$_{CP}$ nachdem er t_1 gelesen hat] ?
what is Fritz fallen asleep after he read has

(26) *Adjunct Condition^d*:
Movement must not take place from an XP that has been merged without a deletion of selectional features.

(27) ADJ-CON^r ("Adjunct Condition", representational version):
A trace must not be separated by a non-selected XP from its antecedent.

Approach no. 1: Gen:
Pesetsky (1997; 1998) emphasizes that certain sentences may be ungrammatical not because they are classified as suboptimal in the H-Eval part of the grammar, but because they cannot be generated by Gen in the first place. Thus, a constraint like (27) might be part of Gen.

Approach no. 2: semantic illformedness:
It is suggested in Grimshaw (1994) and Müller (1997) that certain optimal candidates may have properties that make them inaccessible for other domains of the language faculty like, e.g., semantic interpretation. ADJ-CON might be part of H-Eval, but ranked higher than OP-SPEC. On this view, (28) could block (27) as suboptimal; but this optimal candidate would be uninterpretable (indicated by #) and, hence, unusable.

(28) *Semantic illformedness*:

#Fritz ist eingeschlafen [$_{CP}$ nachdem er was$_1$ gelesen hat] ?
Fritz is fallen asleep after he what read has

Note:
These two approaches have in common that they allow the possibility that absolute ungrammaticality is not located in the H-Eval component of grammar; but in a component that precedes (Gen) or follows (interpretation) optimization. If, however, H-Eval is to be held responsible for the ungrammaticality of (25), there must be a competing candidate with a better constraint profile that blocks it.

Approach no. 3: last resort (fails):
A priori, this might be a candidate that employs a resumptive pronoun strategy which is only legitimate in this context, as a last resort. If this were so, the ineffability problem would be spurious in the case at hand. However, (29) shows that the resumptive pronoun strategy is not an option in German (a constraint like RES must outrank ADJ-CON and other locality constraints in German):

(29) *Last resort*:

*Was$_1$ ist Fritz eingeschlafen [$_{CP}$ nachdem er es$_1$ gelesen hat] ?
what is Fritz fallen asleep after he it read has

Approach no. 4: null parse:
What, then, could the optimal candidate blocking (25) look like? Following Prince & Smolensky (1993), Ackema & Neeleman (1998) propose that the empty candidate Ø (the "null parse") is part of every candidate set. This candidate violates the constraint in (30), which is typically ranked high.

(30) *Ø^r ("Avoid Null Parse"):

Ø is prohibited.

Note:
Constraints that are ranked higher than *Ø in effect become inviolable (given that there is no other constraint except *Ø that Ø can violate). In this sense, *Ø introduces a dividing line into rankings. Thus, if both ADJ-CON and the constraint that triggers *wh*-movement (e.g., OP-SPEC) outrank *Ø, adjunct islands become inviolable. This is shown in table T_{11}.

T_{11}: Adjunct islands and the null parse

Candidates	ADJ-CON	OP-SPEC	*Ø
C_1: was_1 ... [$_{CP}$ nachdem er t_1 V]	*!		
C_2: – ... [$_{CP}$ nachdem er was_1 V]		*!	
☞C_3: Ø			*

Approach no. 5: neutralization:
A final possibility to be discussed here is the *neutralization* approach to absolute ungrammaticality in syntax. Such an approach has been adopted by Legendre, Smolensky & Wilson (1998), Schmid (1998), Baković & Keer (2001), and Wilson (2001), among others. For the present case, a neutralization analysis might posit that the optimal candidate blocking (25) is (31).

(31) *Neutralization*:

Fritz ist eingeschlafen [$_{CP}$ nachdem er was_1 gelesen hat]
Fritz is fallen asleep after he something read has

Note:
The crucial difference to (28) is that *was*$_1$ is turned into an indefinite pronoun, and the matrix $C_{[+wh]}$ is turned into a $C_{[-wh]}$. Thus, there is a feature change from [+wh] in (25) to [–wh] in (31), and the sentence is interpreted as declarative, rather than a as question. If (31) is to block (25) as suboptimal, this presupposes that candidates that differ in their *wh*-feature specification can compete. But then, the problem arises that we would also wrongly expect one of the sentences in (32) to block the other.

(32) *A problem?*

a. Was_1 hat er t_1 gelesen ?
what has he read

b. Er hat was_1 gelesen
he has something read

The solution:
The neutralization approach solves this problem as follows. The [±wh]-specification is unambiguously specified in the input; an input with a [+wh] specification on some item and a minimally different input with a [–wh] specification count as different, and define different candidate sets. The important assumption is that there is a faithfulness constraint that

demands preservation of the [±wh] feature specification in the output:

(33) FAITH[WH]^g:
The output value of [±wh] is the same as the input value.

Analysis:
Suppose now that ADJ-CON and OP-SPEC are ranked higher than FAITH[WH]. Then, (31) will have a better constraint profile than (25) both in the competition that has a [–wh] specification in the input, and in the competition that has a [+wh] specification in the input. Thus, there is a "neutralization" of different input specifications in the output. This is shown in tables T_{12} and T_{13}.

T_{12}: Adjunct islands and neutralization; [–wh] in the input

Candidates	ADJ-CON	OP-SPEC	FAITH[WH]
C_1: $was_{1[+w]}$ ... [$_{CP}$ nachdem er t_1 V]	*!		*
C_2: – ... [$_{CP}$ nachdem er $was_{1[+w]}$ V]		*!	*
☞C_3: – ... [$_{CP}$ nachdem er $was_{1[-w]}$ V]			

T_{13}: Adjunct islands and neutralization; [+wh] in the input

Candidates	ADJ-CON	OP-SPEC	FAITH[WH]
C_1: $was_{1[+w]}$ ... [$_{CP}$ nachdem er t_1 V]	*!		
C_2: – ... [$_{CP}$ nachdem er $was_{1[+w]}$ V]		*!	
☞C_3: – ... [$_{CP}$ nachdem er $was_{1[-w]}$ V]			*

Note:
In transparent contexts, where movement may occur without a violation of a high-ranked locality constraint like ADJ-CON (cf. (32)), FAITH[WH] violations become fatal, and the candidate that maintains the [±wh] specification of the input emerges as optimal.

Conclusion:
Of the four working approaches to absolute ungrammaticality discussed here (Gen, semantic illformedness, null parse, neutralization), the neutralization approach is arguably the most elegant one. Still, it is not without problems. One conspicuous peculiarity is that neutralization creates massive derivational ambiguity. A well-formed sentence like (31) can have different "histories," being an optimal candidate in two candidate sets with different inputs. This vacuous ambiguity may be considered problematic from the point of view of language acquisition and parsing; and it can only be avoided by additional meta-optimization procedures that compare the competitions in T_{12} and T_{13}; cf. the notion of *input optimization* in Prince & Smolensky (1993) (called *lexicon optimization* in phonology).

6.3. Conclusion

Note:
This does not exhaust the list of open issues that are currently under debate in optimality-theoretic syntax. Here are some others:
(i) Optionality
(ii) Degrees of Grammaticality
(iii) Cumulativity
(iv) Parametrization
(v) Serial vs. parallel optimization

Outlook:
Do optimality-theoretic and minimalist approaches differ? Yes, they do, but not as much as one might think at first sight, and in comparing these two approaches, it is important to identify issues that are orthogonal (e.g., derivational vs. representational constraints/candidates, and the issue of transderivational/translocal constraints).