Phonology **31** (2014). Supplementary materials

Review article

Bruce Tesar (2014). *Output-driven phonology: theory and learning*. (Cambridge Studies in Linguistics 139.) Cambridge: Cambridge University Press. Pp. xix + 415.

Giorgio Magri
CNRS, Université Paris 8 and University of Utrecht

Supplementary materials

Appendix A: Merchant's (2008) CPR

This appendix sketches Merchant's (2008) CPR algorithm. In §1, I explain how CPR adheres to the architecture in (28), only with a more abstract representation of the current partial ranking information in terms of ERCs. I then review CPR's subroutines for extracting ranking and lexical information. I start from the easier case, where they are applied to a single paradigmatic entry (§3 and §5), and then turn to the extension to contrast pairs (§4 and §6).

1 ERCs and CPR's architecture

An ELEMENTARY RANKING CONDITION (ERC) is an assignment of one of the three symbols L, W and *e* to each of the constraints in the constraint set (Prince 2002). An ERC is usually represented as a row of these symbols, with the constraints annotated on top. To illustrate, an ERC for the constraint set (5) of the Paka typology is provided in (35). An ERC MATRIX is a finite collection of ERCs, stacked one on top of the other (the order does not matter). An arbitrary ERC matrix is denoted by $\mathfrak{R}$. Given two ERC matrixes $\mathfrak{R}_1$ and $\mathfrak{R}_2$, $\mathfrak{R}_1 + \mathfrak{R}_2$ denotes the ERC matrix obtained by stacking one on top of the other (the order does not matter).

(35)

Id[str]	Id[length]	Main-L	Main-R	NoLong	WSP
[*e*	W	W |	L	L	*e*]

A constraint ranking is CONSISTENT with an ERC provided it satisfies condition (36); it is consistent with an ERC matrix provided it is consistent with each of its rows. An ERC matrix ENTAILS another ERC matrix provided every ranking consistent with the former is also consistent with the latter. Thus $\mathfrak{R}_1 + \mathfrak{R}_2$ entails both $\mathfrak{R}_1$ and $\mathfrak{R}_2$.

(36) At least one constraint which has a W in the ERC under consideration is ranked above every constraint which instead has an L (constraints which have an *e* play no role in the consistency condition).

The phonological underpinning of these notions is as follows. Consider the mapping (**Aa**, **Xx**) of an underlying form **Aa** to a surface form **Xx**, which is thus construed as the winner. For any other loser candidate **Yy**, consider the ERC defined in (37). Stack all the ERCs thus constructed for all loser candidates one on top of the other into an ERC matrix, denoted by $\mathfrak{R}$(**Aa**, **Xx**). Then the OT grammar corresponding to a ranking maps the underlying form **Aa** to the surface form **Xx** if and only if that ranking is consistent with the ERC matrix $\mathfrak{R}$(**Aa**, **Xx**) corresponding to that mapping.

(37) a. A constraint has a W provided it prefers the winner mapping (**Aa**, **Xx**) to the loser mapping (**Aa**, **Yy**), i.e. it assigns fewer violations to the former mapping than to the latter.
b. A constraint has an L provided it instead prefers the loser mapping (**Aa**, **Yy**) to the winner mapping (**Aa**, **Xx**), i.e. it assigns fewer violations to the former mapping than to the latter.
c. A constraint has an *e* provided it has no preference for one of the two mappings (**Aa**, **Xx**) and (**Aa**, **Yy**), i.e. it assigns the same number of violations to both.

To illustrate, consider the Paka typology in §2.1 of the paper, focusing on the mapping (/ˈpaːˈka/, [ˈpaːka]) of the underlying form /ˈpaːˈka/ to the surface form [ˈpaːka]. The ERC matrix $\mathfrak{R}$(/ˈpaːˈka/, [ˈpaːka]) corresponding to this mapping is provided in (38). It consists of seven ERCs, because there are seven loser candidates. For convenience, each ERC is annotated on the left with the corresponding underlying, winner and loser forms (loser forms are struck through). The topmost ERC in (38) says that the winner mapping (/ˈpaːˈka/, [ˈpaːka]) beats the loser mapping (/ˈpaːˈka/, [~~paˈka~~]), provided at least one of the two winner-preferring constraints IDENT[length] or MAIN-L is ranked above the two loser-preferring constraints MAIN-R and NOLONG.

(38)

	ID [str]	ID [length]	MAIN-L	MAIN-R	NO LONG	WSP
/ˈpaːˈka/, [ˈpaːka], [~~paˈka~~]		W	W	L	L	
/ˈpaːˈka/, [ˈpaːka], [~~paˈkaː~~]		W	W	L		
/ˈpaːˈka/, [ˈpaːka], [~~paːˈka~~]			W	L		W
/ˈpaːˈka/, [ˈpaːka], [~~paːˈkaː~~]		W	W	L	W	W
/ˈpaːˈka/, [ˈpaːka], [~~ˈpaka~~]		W			L	
/ˈpaːˈka/, [ˈpaːka], [~~ˈpakaː~~]		W				W
/ˈpaːˈka/, [ˈpaːka], [~~ˈpaːkaː~~]		W			W	W

Grammatical information was represented in §4.1.2 as a (possibly partial) collection G of mappings (**Aa**, **Xx**) of an underlying form **Aa** to a surface form **Xx**. Each of these mappings can be equivalently represented with the corresponding ERC block $\mathfrak{R}$(**Aa**, **Xx**). And the grammatical information G can thus be represented with the ERC matrix obtained by stacking all these blocks one of top of the other. Merchant's CPR adopts the architecture in (28), the only difference being that the current partial grammatical information is represented not as a partial grammar G but as an ERC matrix $\mathfrak{R}$. This makes the algorithm more flexible in the type of partial grammatical information it maintains: any (partial) grammar G can be represented through the corresponding ERC matrix, while not just any ERC matrix $\mathfrak{R}$ corresponds to some partial collection of mappings. This additional flexibility in the data structure maintained by the algorithm is crucial in order to define the subroutine for extracting ranking information, as explained in the next section.

2 The join

The JOIN of some ERC matrices $\mathfrak{R}_1, \ldots, \mathfrak{R}_n$ is any ERC matrix which satisfies the two conditions in (39). Any such ERC matrix will be denoted by $\mathcal{J}(\mathfrak{R}_1, \ldots, \mathfrak{R}_n)$. Condition (39a) says that the join is weaker than (i.e. entailed by) each matrix $\mathfrak{R}_1, \ldots, \mathfrak{R}_n$. Condition (39b) says that, among all ERC matrices which are weaker than (i.e. entailed by) each matrix $\mathfrak{R}_1, \ldots, \mathfrak{R}_n$, the join is a strongest one (i.e the one which entails all the others). In other words, condition (39a) says that the join captures some of the ranking information shared by the matrices $\mathfrak{R}_1, \ldots, \mathfrak{R}_n$, and condition (39b) says there is no shared ranking information which the join does not capture.

(39) a. Each of the ERC matrices $\mathfrak{R}_1, \ldots, \mathfrak{R}_n$ entails their join.
b. If each of the ERC matrices $\mathfrak{R}_1, \ldots, \mathfrak{R}_n$ entails some ERC matrix $\mathfrak{R}$, their join entails $\mathfrak{R}$ as well.

Lemma 3 collects some properties of the join which will be used in the rest of this appendix.

LEMMA 3

(i) Suppose that one of the ERC matrices $\mathfrak{R}_1, \mathfrak{R}_2, \ldots, R_n$ is entailed by each of the others. For concreteness, assume that it is $\mathfrak{R}_1$ which is entailed by each of the other ERC matrices $\mathfrak{R}_2, \ldots, R_n$. Then $\mathfrak{R}_1$ is the join of $\mathfrak{R}_1, \mathfrak{R}_2, \ldots, R_n$.

(ii) Given the ERC matrices $\mathfrak{R}_1, \mathfrak{R}_2, \mathfrak{R}_3, \ldots, R_n$, replace two of them with the ERC matrix obtained by stacking the two matrices one on top of the other. For concreteness, replace $\mathfrak{R}_1$ and $\mathfrak{R}_2$ with $\mathfrak{R}_1 + \mathfrak{R}_2$. Then the join of the $n-1$ ERC matrices $\mathfrak{R}_1 + \mathfrak{R}_2, \mathfrak{R}_3, \ldots, R_n$ entails the join of the original n ERC matrices $\mathfrak{R}_1, \mathfrak{R}_2, \mathfrak{R}_3, \ldots, R_n$, while the reverse does not in general hold.

(iii) Given four ERC matrices $\mathfrak{R}_1$, $\mathfrak{R}_2$, $\widehat{\mathfrak{R}}_1$, $\widehat{\mathfrak{R}}_2$, consider the ERC matrix $\mathcal{J}(\mathfrak{R}_1, \mathfrak{R}_2) + \mathcal{J}(\widehat{\mathfrak{R}}_1, \widehat{\mathfrak{R}}_2)$ obtained by stacking one on top of the other the join of $\mathfrak{R}_1$ and $\mathfrak{R}_2$ and the join of $\widehat{\mathfrak{R}}_1$ and $\widehat{\mathfrak{R}}_2$. Then $\mathcal{J}(\mathfrak{R}_1, \mathfrak{R}_2) + \mathcal{J}(\widehat{\mathfrak{R}}_1, \widehat{\mathfrak{R}}_2)$ coincides with the join of the four ERC matrices $\mathfrak{R}_1 + \widehat{\mathfrak{R}}_1$, $\mathfrak{R}_1 + \widehat{\mathfrak{R}}_2$, $\mathfrak{R}_2 + \widehat{\mathfrak{R}}_1$, $\mathfrak{R}_2 + \widehat{\mathfrak{R}}_2$.

Proof. An ERC matrix $\mathfrak{I}$ is the join of two ERC matrixes $\mathfrak{R}_1$ and $\mathfrak{R}_2$ if and only if the set of rankings consistent with $\mathfrak{I}$ is the union of the two sets of rankings consistent with $\mathfrak{R}_1$ and $\mathfrak{R}_2$. Furthermore, the set of rankings consistent with $\mathfrak{R}_1 + \mathfrak{R}_2$ is the intersection of the two sets of rankings consistent with $\mathfrak{R}_1$ and $\mathfrak{R}_2$. Claims (i)–(iii) then follow from elementary set-theoretic manipulations on the sets of consistent rankings.

3 A subroutine for extracting ranking information through the join

3.1 *Description of the subroutine*. The subroutine in (40) takes as input some ranking information in the form of an ERC matrix $\mathfrak{R}$, some lexical information in the form of a (possibly partial) lexicon *Lex* and a paradigm π (line 1). It extends the ranking information encoded in the input ERC matrix $\mathfrak{R}$ by adding to that input ERC matrix some additional ERCs as follows. The subroutine considers a certain meaning combination Mm (line 2), and it reads off the paradigm π the surface realisation **Xx** of that meaning combination Mm (line 3). The subroutine then constructs the set Γ of all underlying concatenations **Aa** which are consistent with the lexicon *Lex* as underlying forms for that meaning combination Mm (line 4). This means that if *Lex* sets a feature for the root M or the suffix m to a certain value, then the root morpheme **A** or the suffix morpheme **a** of each underlying form **Aa** in Γ has the feature set to that value. Any of these underlying forms in Γ could be the actual underlying form according to the partial lexical information encoded by *Lex*. In other words, the target grammar could enforce any of the mappings (**Aa**, **Xx**) of any underlying form **Aa** in Γ to the surface form **Xx**. Thus the best the subroutine can do is to extract any ranking information shared by all these mappings. This is done by computing the join of all ERC blocks $\mathfrak{R}$(**Aa**, **Xx**) corresponding to all mappings (**Aa**, **Xx**) of all underlying forms **Aa** in Γ to the surface form **Xx** (line 5). The input ERC matrix $\mathfrak{R}$ is updated by adding this join to it (line 6). The ERC matrix thus updated is returned (line 7).

(40)
1. **Require:** an ERC matrix $\mathfrak{R}$, a (possibly partial) lexicon *Lex*, a paradigm π
2. **Require:** a meaning combination Mm in the paradigm π
3. read the surface realisation **Xx** of the meaning combination Mm off the paradigm π
4. construct the set Γ of all concatenations **Aa** consistent with *Lex* as underlying forms for Mm

5. compute the join of the ERC matrices $\mathfrak{R}$(**Aa**, **Xx**) for all underlying forms **Aa** in Γ
6. add this join to the ERC matrix $\mathfrak{R}$
7. **Return:** the updated ERC matrix $\mathfrak{R}$

The join of the ERC blocks $\mathfrak{R}$(**Aa**, **Xx**) computed at line 5 is an ERC-theoretic construct which might very well not correspond, in the sense of (37), to any mapping of an underlying form to a surface form. For this reason, CPR needs the additional flexibility of representing the current (partial) grammatical/ranking information as an ERC matrix rather than as a (partial) grammar. Appendix B: §1 shows that output-drivenness in effect guarantees that the join computed in line 5 always corresponds to the mapping of a certain underlying form to the surface form **Xx** read at line 3.

3.2 *Analysis of the subroutine.* Lemma 4 provides a straightforward correctness result for Merchant's subroutine (40). Note that the proof only uses property (39a) of the join, not property (39b). In other words, the lemma hinges only on the fact that the join extracts ranking information which is shared by all the ERC blocks $\mathfrak{R}$(**Aa**, **Xx**). Property (39b) is nonetheless crucial, because it ensures that the join does not lack any of this shared ranking information, making subroutine (40) optimal. The crux of Merchant's subroutine is its time complexity. Merchant (2008: ch. 4) develops an elegant algorithm for computing the join which can be used to implement line 5. Unfortunately, his algorithm needs to compute the FUSIONAL CLOSURE of each of the input ERC matrices it has to join. Although there are no results that I am aware of concerning the time complexity of that computation, it is most plausibly intractable in the general case.

LEMMA 4

Merchant's subroutine (40) preserves consistency, in the following sense. Assume that the input paradigm π is consistent with some target lexicon and some target ranking, that the input ERC matrix $\mathfrak{R}$ is consistent with that target ranking, and that the input (possibly partial) lexicon *Lex* is consistent with that target lexicon. The updated ERC matrix returned by subroutine (40) is also consistent with that target ranking.

Proof. Let **Aa** be the underlying form assigned to the meaning combination Mm by the target lexicon. This underlying form **Aa** belongs to the set Γ constructed by the subroutine at line 4, because of the hypothesis that the input lexicon *Lex* is consistent with the target lexicon. Thus the corresponding ERC block $\mathfrak{R}$(**Aa**, **Xx**) is one of the blocks that the subroutine takes the join of at line 5. Since the target ranking is consistent with this ERC block $\mathfrak{R}$(**Aa**, **Xx**), then it is also consistent with the join, by property (39a) of the join. Since the target ranking is also consistent with the input ERC matrix $\mathfrak{R}$, it is then consistent with the updated ERC matrix returned by the subroutine.

4 Extension to contrast pairs

4.1 *Description of the subroutine.* The extension of Merchant's subroutine (40) to contrast pairs is provided in (41). This subroutine takes as input some ranking information in the form of an ERC matrix $\mathfrak{R}$, some lexical information in the form of a (possibly partial) lexicon *Lex*, and a paradigm π (line 1). The subroutine considers some contrast pair (line 2). For concreteness, I am assuming here that the two meaning combinations of the contrast pair share the suffix meaning, i.e. have the shape $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}}\textsc{m}$ (the case where they share the root meaning is handled analogously). The subroutine reads off the paradigm π the surface realisations $\mathbf{Xx}$ and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of those two meaning combinations (line 3). Although $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the surface realisations of the same underlying suffix segment corresponding to the suffix meaning $\textsc{m}$, they can be different, because of morphophonemic alternations in the surface realisation of that underlying suffix triggered by the two different roots M and $\widehat{\mathrm{M}}$. The subroutine then constructs the set Γ of all pairs $(\mathbf{Aa}, \hat{\mathbf{A}}\mathbf{a})$ of underlying concatenations which are consistent with the lexicon *Lex* as underlying forms for the meaning combinations $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}}\textsc{m}$ and furthermore share the underlying suffix segment $\mathbf{a}$ (line 4). The latter requirement encodes the fact that we are focusing on a pair of meaning combinations $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}}\textsc{m}$ which share the suffix meaning $\textsc{m}$. For each pair $(\mathbf{Aa}, \hat{\mathbf{A}}\mathbf{a})$ of underlying concatenations in Γ, the subroutine considers the corresponding ERC blocks $\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx})$ and $\mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}})$, stacks them together into the ERC matrix $\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}})$ and constructs the join of all ERC matrices thus obtained (line 5). The input ERC matrix $\mathfrak{R}$ is updated by adding this join to it (line 6). The ERC matrix thus updated is then returned (line 7).

(41)
1. **Require:** an ERC matrix $\mathfrak{R}$, a (possibly partial) lexicon *Lex*, a paradigm π
2. **Require:** a pair $(\mathrm{M}\textsc{m}, \widehat{\mathrm{M}}\textsc{m})$ of meaning concatenations which share, say, the suffix meaning $\textsc{m}$
3. read the surface realisation $\mathbf{Xx}$ and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}}\textsc{m}$ off the paradigm π
4. construct the set Γ of pairs $(\mathbf{Aa}, \hat{\mathbf{A}}\mathbf{a})$ of concatenations sharing the suffix morpheme such that
   - **Aa** is consistent with the lexicon *Lex* as an underlying form for $\mathrm{M}\textsc{m}$
   - $\hat{\mathbf{A}}\mathbf{a}$ is consistent with the lexicon *Lex* as an underlying form for $\widehat{\mathrm{M}}\textsc{m}$
5. compute the join of the ERC matrices $\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}})$ for all pairs $(\mathbf{Aa}, \hat{\mathbf{A}}\mathbf{a})$ in Γ
6. add this join to the ERC matrix $\mathfrak{R}$
7. **Return:** the updated ERC matrix $\mathfrak{R}$

There is a subtlety which should be pointed out here. At line 5, the subroutine considers the join in (42a), not the one in (42b). The difference is that in (42a) we sum together the two ERC blocks $\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx})$ and $\mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}})$ into $\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}})$ before taking the join. This is indeed appropriate, because the join in (42a) is stronger than the one in (42b), by Lemma 3.ii.

(42) a. $\mathcal{J}\{\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}}) \mid (\mathbf{Aa}, \hat{\mathbf{A}}\mathbf{a}) \in \Gamma\}$
b. $\mathcal{J}\{\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx}), \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}, \hat{\mathbf{X}}\hat{\mathbf{x}}) \mid (\mathbf{Aa}, \hat{\mathbf{A}}\mathbf{a}) \in \Gamma\}$

4.2 *Analysis of the subroutine*. The Correctness Lemma 4 trivially extends from the original subroutine (40) for single meaning combinations to the variant (41) for contrast pairs. The inefficiency of (40) for single meaning combinations is of course aggravated in (41) for contrast pairs, as the latter requires taking the join of a larger number of larger ERC matrices. Yet subroutine (41) applied to the contrast pair $(\mathrm{M}\textsc{m}, \widehat{\mathrm{M}\textsc{m}})$ is able to extract more ranking information than the double application of the original subroutine (40) to the two separate meaning combinations $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}\textsc{m}}$. To see this, suppose that the root meaning M admits a unique underlying form $\mathbf{A}$ consistent with *Lex*, the root meaning $\widehat{\mathrm{M}}$ admits a unique underlying form $\hat{\mathbf{A}}$ and the suffix meaning $\textsc{m}$ admits only two underlying forms, $\mathbf{a}_1$ and $\mathbf{a}_2$. Suppose we first apply the original subroutine (40) to the meaning combination $\mathrm{M}\textsc{m}$. At line 4, it constructs the set Γ consisting of the two underlying forms $\mathbf{Aa}_1$ and $\mathbf{Aa}_2$. At line 5, it computes the join $\mathcal{J}(\mathfrak{R}_1, \mathfrak{R}_2)$ of the two ERC blocks $\mathfrak{R}_1 = \mathfrak{R}(\mathbf{Aa}_1, \mathbf{Xx})$ and $\mathfrak{R}_2 = \mathfrak{R}(\mathbf{Aa}_2, \mathbf{Xx})$. Suppose we next apply the original subroutine (40) to the meaning combination $\widehat{\mathrm{M}\textsc{m}}$. At line 4, it constructs the set Γ, consisting of the two underlying forms $\hat{\mathbf{A}}\mathbf{a}_1$ and $\hat{\mathbf{A}}\mathbf{a}_2$. At line 5, it computes the join $\mathcal{J}(\hat{\mathfrak{R}}_1, \hat{\mathfrak{R}}_2)$ of the two ERC blocks $\hat{\mathfrak{R}}_1 = \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}_1, \hat{\mathbf{X}}\hat{\mathbf{x}})$ and $\hat{\mathfrak{R}}_2 = \mathfrak{R}(\hat{\mathbf{A}}\mathbf{a}_2, \hat{\mathbf{X}}\hat{\mathbf{x}})$. Ultimately, the two consecutive applications of the original subroutine (40) to the two meaning combinations $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}\textsc{m}}$ extract the ranking information captured by the sum of the two joins in (43).

(43) $\mathcal{J}(\mathfrak{R}_1, \mathfrak{R}_2) + \mathcal{J}(\hat{\mathfrak{R}}_1, \hat{\mathfrak{R}}_2)$

Lemma 3.iii ensures that (43) is equivalent to the join in (44). The four ERC matrices being joined here correspond to all four combinations of underlying forms for the two meaning combinations $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}\textsc{m}}$. In particular, the learner is also joining together the two ERC blocks $\mathfrak{R}_1 + \hat{\mathfrak{R}}_2$ and $\hat{\mathfrak{R}}_2 + \mathfrak{R}_1$, whereby the shared suffix meaning $\textsc{m}$ is assigned a different underlying form in the case of $\mathrm{M}\textsc{m}$ than in the case of $\widehat{\mathrm{M}\textsc{m}}$. In other words, since the learner is processing the two meaning combinations $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}\textsc{m}}$ separately, it is blind to the fact that these meaning combinations share the suffix meaning $\textsc{m}$, which must therefore have the same underlying form in both cases.

(44) $\mathcal{J}(\mathfrak{R}_1 + \hat{\mathfrak{R}}_1, \mathfrak{R}_1 + \hat{\mathfrak{R}}_2, \mathfrak{R}_2 + \hat{\mathfrak{R}}_1, \mathfrak{R}_2 + \hat{\mathfrak{R}}_2)$

Suppose we instead apply subroutine (41) to the contrast pair (Mᴍ, $\widehat{\mathrm{M}\textsc{m}}$). At line 4, it constructs the set Γ, consisting of the two pairs ($\mathbf{Aa}_1$, $\widehat{\mathbf{A}}\mathbf{a}_1$) and ($\mathbf{Aa}_2$, $\widehat{\mathbf{A}}\mathbf{a}_2$). At line 5, it computes the join in (45) of the two ERC blocks $\mathfrak{R}(\mathbf{Aa}_1, \mathbf{Xx}) + \mathfrak{R}(\widehat{\mathbf{A}}\mathbf{a}_1, \widehat{\mathbf{X}}\widehat{\mathbf{x}}) = \mathfrak{R}_1 + \widehat{\mathfrak{R}}_1$ and $\mathfrak{R}(\mathbf{Aa}_2, \mathbf{Xx}) + \mathfrak{R}(\widehat{\mathbf{A}}\mathbf{a}_2, \widehat{\mathbf{X}}\widehat{\mathbf{x}}) = \mathfrak{R}_2 + \widehat{\mathfrak{R}}_2$.

(45) $\mathcal{J}(\mathfrak{R}_1 + \widehat{\mathfrak{R}}_1, \mathfrak{R}_2 + \widehat{\mathfrak{R}}_2)$

The join in (45) is stronger than the one in (44), because it joins fewer matrices (the smaller the number of matrices, the larger the amount of shared information for the join to extract). This is because the two blocks $\mathfrak{R}_1 + \widehat{\mathfrak{R}}_2$ and $\mathfrak{R}_2 + \widehat{\mathfrak{R}}_1$ have disappeared from the join. This is what we want, as they correspond to the unreasonable assumption that the suffix ᴍ has different underlying forms in the two meaning combinations Mᴍ and $\widehat{\mathrm{M}\textsc{m}}$. In other words, since the learner is processing the two meaning combinations Mᴍ and $\widehat{\mathrm{M}\textsc{m}}$ together as a contrast pair, it is able to take into account the fact that these meaning combinations share the suffix meaning ᴍ, which must therefore have the same underlying form in the two cases.

5 A subroutine for extracting lexical information through inconsistency detection

5.1 *Description of the subroutine.* Subroutine (46) takes as input some lexical information in the form of a partial lexicon *Lex*, some ranking information in the form of a (possibly partial) grammar G (or an ERC matrix $\mathfrak{R}$) and a paradigm π (line 1). The subroutine then considers a certain meaning combination Mᴍ with some feature unset according to the input partial lexicon *Lex* for either the root M or the suffix ᴍ (line 2). The goal of the subroutine is to try to extend the input partial lexicon by setting some of these unset features. To this end, the subroutine reads off the paradigm π the surface realisation **Xx** of the meaning combination Mᴍ (line 3). It constructs the set Γ of all underlying concatenations **Aa** which are consistent with both the input lexical information *Lex* and the input ranking information captured by the grammar G (or the ERC matrix $\mathfrak{R}$) (line 4). Consistency with the input lexical information means that, if *Lex* sets a feature for the root M (or the suffix ᴍ) to a certain value, then the root morpheme **A** (or the suffix morpheme **a**) of each underlying form **Aa** in Γ has the feature set to that value. Consistency with the input ranking information means that the input grammar G (or the input ERC matrix $\mathfrak{R}$) is consistent with the mapping (**Aa**, **Xx**) of each such underlying form **Aa** in Γ to the surface form **Xx**. The subroutine then looks for a feature which is unset by *Lex* for the root meaning M (or the suffix meaning ᴍ), such that each root morpheme **A** (or each suffix morpheme **a**) has the same value for that feature in all underlying concatenations **Aa** in Γ (line 5). For any such feature, it can be concluded that the opposite value is inconsistent with the available lexical and ranking information. The subroutine thus updates the partial lexicon *Lex* by setting that feature to be equal to its constant value for the root meaning M (or for the suffix meaning m) (line 6). The lexicon thus updated is then returned (line 8).

(46)
1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar *G* (or an ERC matrix $\mathfrak{R}$), a paradigm π
2. **Require:** a meaning combination Mm in the paradigm π
3. read the surface realisation **Xx** of the meaning combination Mm off the paradigm π
4. construct the set Γ of all the underlying concatenations **Aa** such that:
   - **Aa** is consistent with the lexicon *Lex* as an underlying form for Mm
   - the mapping (**Aa**, **Xx**) is consistent with the grammar *G* (or the ERC matrix $\mathfrak{R}$)
5. **if** some feature unset for the root M (or for the suffix m) has a constant value over Γ for M (for m) **then**
6.   update the lexicon *Lex* by setting the feature to that value for the root M (or for the suffix m)
7. **end if**
8. **Return:** the updated lexicon *Lex*

This subroutine is called INCONSISTENCY DETECTION (cf. Kager 1999, Tesar *et al*. 2003, Tesar 2006), because it sets a certain feature to a certain value based on the detection at line 6 of the fact that any other value for that feature would be inconsistent with the available lexical and ranking information.

5.2 *Analysis of the subroutine*. Lemma 5 provides a straightforward correctness result for inconsistency detection (46). Its time complexity is controlled by the size of the set Γ of consistent underlying forms constructed at line 4. In the worst case (when both the input lexicon and the input ranking information are empty), Γ consists of all underlying forms, i.e. all concatenations of root and suffix morphemes. In this case, the size of Γ grows polynomially in the number of morphemes and thus exponentially in the number of features. In conclusion, subroutine (46) is efficient relative to the generous notion of efficiency defined in (14), which measures time complexity in terms of the number of morphemes, but is not efficient relative to the more demanding notion of efficiency defined in (15), which measures time complexity in terms of the number of features.

LEMMA 5
Inconsistency detection (46) preserves consistency, in the following sense. Assume that the input paradigm π is consistent with some target lexicon and some target grammar, that the input (possibly partial) grammar *G* (or ERC matrix $\mathfrak{R}$) is consistent with the target grammar

and that the input partial lexicon *Lex* is consistent with the target lexicon. The updated lexicon returned by subroutine (46) is also consistent with that target lexicon.

Proof. Assume by contradiction that the lemma is false. This means that the subroutine has set some feature φ for, say, the root meaning M to a value which is different from the value assigned by the target lexicon. For concreteness, assume that the values assigned by the subroutine and the target lexicon are + and − respectively. Thus , the target lexicon assigns to the meaning combination Mm an underlying form **Aa**, which has feature φ set to the value − in the root **A**. This underlying form **Aa** is consistent with the partial lexicon *Lex* (because it is consistent with the target lexicon, which is in turn consistent with *Lex*). Furthermore, the mapping (**Aa**, **Xx**) of this underlying form **Aa** to the surface form **Xx** is consistent with the input grammar *G* (because the target ranking is consistent with both). In other words, this underlying form **Aa** satisfies the two conditions stated by the two subclauses in line 4 and thus belongs to the set Γ constructed by the subroutine. Since this underlying form **Aa** has feature φ set equal to − for the root **A**, then the subroutine cannot have set that feature to the opposite value + at line 6.

6 Extension to contrast pairs

6.1 *Description of the subroutine*. The extension of inconsistency detection (46) to contrast pairs is provided in (47). This subroutine takes as input some lexical information in the form of a partial lexicon *Lex*, some ranking information in the form of a (possibly partial) grammar *G* (or an ERC matrix $\mathfrak{R}$) and a paradigm π (line 1). The subroutine considers a particular contrast pair (line 2). For concreteness, I am assuming here that the two meaning combinations of the contrast pair share the suffix meaning, namely have the shape Mm and $\widehat{\text{M}}$m (the case where they share the root meaning is handled analogously). The subroutine reads off the paradigm π the surface realisations **Xx** and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of those two meaning combinations (line 3). It constructs the set Γ of all pairs (**Aa**, $\hat{\mathbf{A}}$**a**) of underlying concatenations which are consistent with both the input lexical information *Lex* and the input ranking information *G* (or $\mathfrak{R}$) and furthermore share the underlying suffix segment **a** (line 4). The subroutine then looks for a feature which is unset by *Lex* for the root meaning M (or the root meaning $\widehat{\text{M}}$ or the suffix meaning m), such that each underlying root morpheme **A** (or each underlying root morpheme $\hat{\mathbf{A}}$ or each underlying suffix morpheme **a**) has the same value for that feature in all pairs (**Aa**, $\hat{\mathbf{A}}$**a**) in Γ (line 5). For any such feature, the subroutine updates the partial lexicon *Lex* by setting the feature equal to its constant value for the root meaning M (or for the root meaning $\widehat{\text{M}}$ or for the suffix meaning m) (line 6). The lexicon thus updated is then returned (line 8).

(47) 1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar G (or an ERC matrix $\Re$), a paradigm π

2. **Require:** a pair (Mᴍ, $\widehat{\text{Mᴍ}}$) of meaning concatenations which share, say, the suffix meaning ᴍ

3. read the surface realisations **Xx** and **X̂x̂** of Mᴍ and $\widehat{\text{Mᴍ}}$ off the paradigm π

4. construct the set Γ of pairs (**Aa**, **Âa**) of concatenations sharing the suffix morpheme such that
   - **Aa** is consistent with the lexicon *Lex* as an underlying form for Mᴍ
   - the mapping (**Aa**, **Xx**) is consistent with the grammar G (or the ERC matrix $\Re$)
   - **Âa** is consistent with the lexicon *Lex* as an underlying form for $\widehat{\text{Mᴍ}}$
   - the mapping (**Âa**, **X̂x̂**) is consistent with the grammar G

5. **if** some unset feature has a constant value over Γ for any of the three meanings M, $\widehat{\text{M}}$ or ᴍ **then**

6.   update the lexicon *Lex* by setting the feature to the constant value for that meaning

7. **end if**

8. **Return:** the updated lexicon *Lex*

6.2 *Analysis of the subroutine*. The Correctness Lemma 5 trivially extends from the original subroutine (46) for single meaning combinations to the variant (47) for contrast pairs. Furthermore, the variant for contrast pairs remains as efficient as the original version for single meaning combinations, when efficiency is generously measured in terms of the number of morphemes (but again, not when it is measured in terms of the number of features). Yet a number of authors (including Alderete *et al.* 2005 and Merchant & Tesar 2008) have pointed out that (47), applied to the contrast pair (Mᴍ, $\widehat{\text{Mᴍ}}$), can extract more lexical information than the double application of the original subroutine (46) to the two separate meaning combinations Mᴍ and $\widehat{\text{Mᴍ}}$. To see this, consider one of the two meaning combinations of the contrast pair, say Mᴍ. The original subroutine considers all underlying forms **Aa** for Mᴍ which are consistent with the input lexical and ranking information, and only sets those features which are constant across all these underlying forms. If the number of consistent underlying forms is large, the number of features which are constant across them is small, and the subroutine thus extracts little lexical information. (47) might instead be able to extract more lexical information, because it looks for features which are constant over a smaller set of underlying forms for Mᴍ, namely those underlying forms **Aa** which are not only consistent with the input lexical and ranking information, but also use a suffix morpheme **a** which can be combined with a root morpheme **Â** into an underlying form **Âa** for $\widehat{\text{Mᴍ}}$ which is also consistent with the input lexical and ranking information.

Appendix B: Equivalence between Merchant's and Tesar's subroutines

1 Subroutines for extracting ranking information from a single meaning combination

I denote Tesar's subroutine (29) for the extraction of ranking information from a single meaning combination as $\mathrm{ERI_T}$, and Merchant's subroutine (40) as $\mathrm{ERI_M}$. Lemma 6 states that, if the underlying typology is output-driven, then $\mathrm{ERI_T}$ and $\mathrm{ERI_M}$ are equivalent, because they update the current ranking/grammatical information with the 'same' information. This equivalence is striking, given that the time complexity of $\mathrm{ERI_T}$ is only linear in the number of features, while the time complexity of $\mathrm{ERI_M}$ is unknown, but probably exponential in both the number of features and the number of morphemes. Output-drivenness thus affords a significant reduction in time complexity. The Correctness Lemma 1 for $\mathrm{ERI_T}$ (which was proven directly in §4.4) can also be made to follow from the Correctness Lemma 4 for $\mathrm{ERI_T}$ through the Equivalence Lemma 6.

LEMMA 6

Suppose that the underlying typology is output-driven and that $\mathrm{ERI_T}$ is fed with a partial grammar *G* and a (possibly partial) lexicon *Lex*. Suppose that $\mathrm{ERI_M}$ is fed with the ERC matrix $\mathfrak{R}$ corresponding to the partial grammar *G* and the same lexicon *Lex*. Then the ERC block corresponding to the mapping (**Bb**, **Xx**) constructed by $\mathrm{ERI_T}$ at line 4 is the join of the ERC blocks $\mathfrak{R}$(**Aa**, **Xx**) corresponding to the underlying forms **Aa** in Γ constructed by $\mathrm{ERI_M}$ at line 5.

Proof. The underlying form **Bb** constructed by $\mathrm{ERI_T}$ at line 4 is consistent with *Lex*, and therefore belongs to the set Γ constructed by $\mathrm{ERI_M}$ at line 4. Consider one of the underlying forms **Aa** in this set Γ. The two underlying forms **Aa** and **Bb** agree in every feature which is set by the input partial lexicon *Lex*. Furthermore, **Bb** and the surface form **Xx** agree in every remaining feature. Thus the underlying form **Bb** is more similar to the surface form **Xx** than is the underlying form **Aa**. Since the underlying typology is output-driven, then every ranking which maps the less similar underlying form **Aa** to **Xx** also maps the more similar underlying form **Bb** to **Xx**. In other words, any ranking consistent with the ERC block $\mathfrak{R}$(**Aa**, **Xx**) is also consistent with the ERC block $\mathfrak{R}$(**Bb**, **Xx**). Lemma 3.i thus ensures that the ERC block $\mathfrak{R}$(**Bb**, **Xx**) is the join of the ERC blocks $\mathfrak{R}$(**Aa**, **Xx**) across all underlying concatenations **Aa** in Γ.

2 Subroutines for extracting lexical information from a single meaning combination

Consider the slight variant (48) of the subroutine (46) for the extraction of lexical information through inconsistency detection. The only difference between the two implementations is the following: Merchant's original subroutine (46) sets any feature which is constant over Γ; the variant in (48) instead focuses right from the beginning on a specific feature φ, which is unset for either the root M or the suffix ᴍ of the meaning combination Mᴍ under consideration. In line 3, I am assuming for concreteness that the feature φ is unset for the root M (the case where it is unset for the suffix ᴍ is handled analogously). The modification is only cosmetic: the original subroutine is equivalent to running this variant for each unset feature.

(48)
1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar G, a paradigm π
2. **Require:** a meaning combination Mᴍ in the paradigm π
3. **Require:** a feature φ unset by *Lex* for either M or ᴍ; for concreteness, say it is unset for M
4. read the surface realisation **Xx** of the meaning combination Mᴍ off the paradigm π
5. construct the set Γ of all the underlying concatenations **Aa** such that
   - **Aa** is consistent with the lexicon *Lex* as an underlying form for Mᴍ
   - the mapping (**Aa**, **Xx**) is consistent with the grammar G
6. **if** all underlying forms **Aa** in Γ have the feature φ set to the same value in the root morpheme **A then**
7. update the lexicon *Lex* by setting the feature φ to that value for the root meaning M
8. **end if**
9. **Return:** the updated lexicon *Lex*

I denote Tesar's subroutine (33) for the extraction of lexical information from a single meaning combination by $\mathrm{ELI}_{\mathrm{T}}$, and (the cosmetic variant of) the original inconsistency detection subroutine (48) by $\mathrm{ELI}_{\mathrm{id}}$. Lemma 6 states that, if the underlying typology is output-driven, $\mathrm{ELI}_{\mathrm{T}}$ and $\mathrm{ELI}_{\mathrm{id}}$ are equivalent, because their if-clauses at line 6 are equivalent. This equivalence is striking, given that the if-clause of $\mathrm{ELI}_{\mathrm{T}}$ can be checked in time linear in the number of features, while the if-clause of $\mathrm{ELI}_{\mathrm{id}}$ requires time exponential in the number of features in the worst case. Output-drivenness has thus afforded a significant reduction in time complexity. The Correctness Lemma 2 for $\mathrm{ELI}_{\mathrm{T}}$ (which was proven directly in §4.5) can also be made to follow from the Correctness Lemma 4 for $\mathrm{ELI}_{\mathrm{id}}$ through the Equivalence Lemma 7.

Lemma 7

Suppose that the underlying typology is output-driven. Then the two if-clauses in line 6 of $\mathrm{ELI}_{\mathrm{T}}$ and $\mathrm{ELI}_{\mathrm{id}}$ are equivalent.

Proof. To start, let me prove that the if-clause in line 6 of $\mathrm{ELI}_{\mathrm{T}}$ entails the if-clause in line 6 of $\mathrm{ELI}_{\mathrm{id}}$. Assume by contradiction that the former if-clause is true while the latter if-clause is false. The hypothesis that the if-clause of $\mathrm{ELI}_{\mathrm{id}}$ is false entails in particular that the set Γ of consistent underlying forms constructed by $\mathrm{ELI}_{\mathrm{id}}$ at line 5 contains an underlying form **Aa** whose root **A** disagrees with **X** relative to the feature φ (indeed, if all the underlying forms **Aa** in Γ had a root **A** which agreed with **X** relative to φ, the if-clause of $\mathrm{ELI}_{\mathrm{id}}$ would hold true). The underlying form **Aa** and the underlying form **Bb** constructed by $\mathrm{ELI}_{\mathrm{T}}$ at line 5 thus agree in the feature φ for the root meaning M, as they both have the opposite value from **X**. Furthermore, they agree in every feature set by the partial lexicon *Lex*, because they are both consistent with *Lex*. Finally, **Bb** agrees with **Xx** for every remaining feature. The underlying form **Bb** is thus more similar to the surface form **Xx** than is the underlying form **Aa**. Since the mapping (**Aa**, **Xx**) is consistent with G (because **Aa** belongs to Γ), there exists a ranking which is consistent with both. Since the grammar corresponding to that ranking is output-driven and maps the less similar underlying form **Aa** to **Xx**, it also maps the more similar underlying form **Bb** to **Xx**. In other words, that ranking is also consistent with the mapping (**Bb**, **Xx**). In conclusion, the mapping (**Bb**, **Xx**) and the grammar G are consistent (because both are consistent with the ranking), contradicting the hypothesis that the if-clause of $\mathrm{ELI}_{\mathrm{T}}$ is true.

Vice versa, let me prove that the if-clause in line 6 of $\mathrm{ELI}_{\mathrm{id}}$ entails the if-clause in line 6 of $\mathrm{ELI}_{\mathrm{T}}$. Assume by contradiction that the former if-clause is true, while the latter if-clause is false. The hypothesis that the if-clause of $\mathrm{ELI}_{\mathrm{T}}$ is false means that the input (partial) grammar G is consistent with the mapping (**Bb**, **Xx**) of the underlying form **Bb** constructed by $\mathrm{ELI}_{\mathrm{T}}$ at line 5 to the surface form **Xx**. Since this underlying form **Bb** is consistent with *Lex*, then **Bb** belongs to the set Γ of consistent underlying forms constructed at line 5 by $\mathrm{ELI}_{\mathrm{id}}$. Since **B** and **X** disagree relative to the feature φ, consider the underlying form $\bar{\mathbf{B}}\mathbf{b}$, identical to **Bb** but for the fact that $\bar{\mathbf{B}}$ and **X** instead agree in the feature φ. Obviously, the underlying form $\bar{\mathbf{B}}\mathbf{b}$ is more similar to the surface form **Xx** than is the underlying form **Bb**. Since (**Bb**, **Xx**) is consistent with G, there exists a ranking which is consistent with both. Since the grammar corresponding to that ranking is output-driven and maps the less similar underlying form **Bb** to **Xx**, then it also maps the more similar underlying form $\bar{\mathbf{B}}\mathbf{b}$ to **Xx**. In other words, that ranking is also consistent with the mapping ($\bar{\mathbf{B}}\mathbf{b}$, **Xx**). This underlying form $\bar{\mathbf{B}}\mathbf{b}$ thus belongs to the set Γ constructed by $\mathrm{ELI}_{\mathrm{id}}$ at line 5.This conclusion contradicts the hypothesis that the if-clause of $\mathrm{ELI}_{\mathrm{id}}$ holds, because it shows that Γ contains two underlying forms **Bb** and $\bar{\mathbf{B}}\mathbf{b}$, such that **B** and $\bar{\mathbf{B}}$ disagree in the feature φ.

3 Subroutines for extracting lexical information from a contrast pair

Tesar's subroutine for extracting lexical information from contrast pairs is more involved than his other two subroutines, for extracting lexical and ranking information from a single meaning combination. For this reason, the latter two subroutines were reviewed in §4.4 and §4.5, while the former subroutine is included in this appendix. Here I start out from Merchant's subroutine in (47) for extracting lexical information from contrast pairs, and work towards Tesar's reformulation, provided below in (52). This seems to me the best way of making sense of the somewhat complex formulation of Tesar's subroutine. I begin by carrying out the same cosmetic change as in the previous section: I replace the original subroutine (47) with (49), which only differs because line 3 focuses on a specific feature φ, unset by the input lexicon *Lex* for one of the three meanings M, $\widehat{\mathrm{M}}$ and $\textsc{m}$ of the contrast pair ($\mathrm{M}\textsc{m}$, $\widehat{\mathrm{M}}\textsc{m}$). For concreteness, I assume it is unset for the root meaning M.

(49)
1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar G, a paradigm π
2. **Require:** a pair ($\mathrm{M}\textsc{m}$, $\widehat{\mathrm{M}}\textsc{m}$) of meaning concatenations which share, say, the suffix meaning $\textsc{m}$
3. **Require:** a feature φ unset by *Lex* for M, $\widehat{\mathrm{M}}$ or $\textsc{m}$; for concreteness, say it is unset for M
4. read the surface realisations $\mathbf{Xx}$ and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}}\textsc{m}$ off the paradigm π
5. construct the set Γ of pairs ($\mathbf{Aa}$, $\hat{\mathbf{A}}\mathbf{a}$) of concatenations sharing the suffix morpheme such that:
   - $\mathbf{Aa}$ is consistent with the lexicon *Lex* as an underlying form for $\mathrm{M}\textsc{m}$
   - the mapping ($\mathbf{Aa}$, $\mathbf{Xx}$) is consistent with the grammar G
   - $\hat{\mathbf{A}}\mathbf{a}$ is consistent with the lexicon *Lex* as an underlying form for $\widehat{\mathrm{M}}\textsc{m}$
   - the mapping ($\hat{\mathbf{A}}\mathbf{a}$, $\hat{\mathbf{X}}\hat{\mathbf{x}}$) is consistent with the grammar G
6. **if** all the pairs ($\mathbf{Aa}$, $\hat{\mathbf{A}}\mathbf{a}$) in Γ have the underlying root $\mathbf{A}$ set to the same value for feature φ **then**
7.   update the lexicon *Lex* by setting feature φ to that constant value for the meaning M
8. **end if**
9. **Return:** the updated lexicon *Lex*

The modification is only cosmetic: the original subroutine (47) is equivalent to running this variant (49) for each feature which is unset for one of the three morphemes M, $\widehat{\mathrm{M}}$ and $\textsc{m}$.

The set Γ constructed by the subroutine (49) at line 5 can equivalently be described as the collection of all pairs ($\mathbf{Aa}$, $\hat{\mathbf{A}}\hat{\mathbf{a}}$) which satisfy the conditions expressed by the four subclauses of line 5 plus the additional condition that the two underlying suffix morphemes **a** and **â** coincide, because they are the underlying form corresponding to the same suffix meaning $\textsc{m}$. The condition that **a** and **â** coincide is in turn equivalent to the condition that they agree for each feature ψ. We could of course relax this condition by requiring that the two underlying suffix morphemes **a** and **â** agree only for certain features, $\psi_1, \ldots, \psi_n$. This intuition leads to the formulation in (50). It differs from (49) only because of the new line 4, which introduces the features $\psi_1, \ldots, \psi_n$, and because of the additional fifth subclause in the definition of the set Γ. Note that line 4 assumes all the features $\psi_1, \ldots, \psi_n$ to be unset by the input partial lexicon *Lex* for the shared suffix $\textsc{m}$, because the condition that both **a** and **â** are consistent with *Lex* as underlying forms for $\textsc{m}$ already entails that they agree on every feature set by *Lex* for $\textsc{m}$.

(50)
1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar G, a paradigm π
2. **Require:** a pair ($\mathrm{M}\textsc{m}$, $\widehat{\mathrm{M}}\textsc{m}$) of meaning concatenations which share, say, the suffix meaning $\textsc{m}$
3. **Require:** a feature φ unset by *Lex* for M, $\widehat{\mathrm{M}}$ or $\textsc{m}$; for concreteness, say it is unset for M
4. **Require:** some features $\psi_1, \ldots, \psi_n$ all unset for the shared suffix meaning $\textsc{m}$ according to *Lex*
5. read the surface realisations **Xx** and $\mathbf{\hat{X}\hat{x}}$ of $\mathrm{M}\textsc{m}$ and $\widehat{\mathrm{M}}\textsc{m}$ off the paradigm π
6. construct the set Γ of pairs (**Aa**, **Ââ**) of concatenations sharing the suffix morpheme such that
   - **Aa** is consistent with the lexicon *Lex* as an underlying form for $\mathrm{M}\textsc{m}$
   - the mapping (**Aa**, **Xx**) is consistent with the grammar G
   - **Âa** is consistent with the lexicon *Lex* as an underlying form for $\widehat{\mathrm{M}}\textsc{m}$
   - the mapping (**Ââ**, $\mathbf{\hat{X}\hat{x}}$) is consistent with the grammar G
   - the two underlying suffix morphemes **a** and **â** agree for every feature $\psi_1, \ldots, \psi_n$
7. **if** all the pairs (**Aa**, **Ââ**) in Γ have the underlying root **A** set to the same value for feature φ **then**
8.   update the lexicon *Lex* by setting feature φ to that constant value for the meaning M
9. **end if**
10. **Return:** the updated lexicon *Lex*

If we consider no ψ features at all in line 4, then (50) effectively considers the two meaning combinations of the contrast pair independently of each other, and thus ends up equivalent to subroutine (46) for extracting lexical information from single meaning combinations, in the sense that the application of subroutine (50) to the contrast pair ($\mathrm{M}\textsc{m}$, $\widehat{\mathrm{M}}\textsc{m}$) is equivalent to two consecutive applications of subroutine (46) to the two meaning combinations Mm and $\widehat{\mathrm{M}}$m one at a time. If instead we consider all unset ψ features at line 4, then (50) effectively considers only pairs of underlying forms for the two meaning combinations which share the underlying suffix, and is therefore equivalent to the original subroutine (49) for extracting lexical information from contrast pairs. In the end, depending on the number of ψ features considered at line 4, (50) provides a whole range of subroutines for extracting lexical information in between the weaker (46) and the stronger (49).

Subroutine (50) can be equivalently restated as (51). Instead of requiring the two suffix morphemes $\mathbf{a}$ and $\hat{\mathbf{a}}$ to coincide for every feature $\psi_1, ..., \psi_n$, we consider all possible combinations $v_1, ..., v_n$ of values of those features at line 6, and we require both $\mathbf{a}$ and $\hat{\mathbf{a}}$ to have those features set equal to those values in the fifth subclause of line 5.

(51)
1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar G, a paradigm π
2. **Require:** a pair (Mm, $\widehat{\mathrm{M}}$m) of meaning concatenations which share, say, the suffix meaning m
3. **Require:** a feature φ unset by *Lex* for M, $\widehat{\mathrm{M}}$ or m; for concreteness, say it is unset for M
4. **Require:** some features $\psi_1, ..., \psi_n$ all unset for the shared suffix meaning m according to *Lex*
5. read the surface realisations $\mathbf{Xx}$ and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of Mm and $\widehat{\mathrm{M}}$m off the paradigm π
6. For any combination $v_1, ..., v_n$ of values of the features $\psi_1, ..., \psi_n$, construct the set $\Gamma(v_1, ..., v_n)$ of pairs ($\mathbf{Aa}$, $\hat{\mathbf{A}}\hat{\mathbf{a}}$) of underlying concatenations such that:
   - $\mathbf{Aa}$ is consistent with the lexicon *Lex* as an underlying form for Mm
   - the mapping ($\mathbf{Aa}$, $\mathbf{Xx}$) is consistent with the grammar G
   - $\hat{\mathbf{A}}\hat{\mathbf{a}}$ is consistent with the lexicon *Lex* as an underlying form for $\widehat{\mathrm{M}}$m
   - the mapping ($\hat{\mathbf{A}}\hat{\mathbf{a}}$, $\hat{\mathbf{X}}\hat{\mathbf{x}}$) is consistent with the grammar G
   - the two underlying suffix morphemes $\mathbf{a}$ and $\hat{\mathbf{a}}$ both have the values $v_1, ..., v_n$ for the features $\psi_1, ..., \psi_n$
7. **if** all the pairs ($\mathbf{Aa}$, $\hat{\mathbf{A}}\hat{\mathbf{a}}$) in all the sets $\Gamma(v_1, ..., v_n)$ have the underlying root $\mathbf{A}$ set to the same value for feature φ **then**
8. update the lexicon *Lex* by setting feature φ to that constant value for the meaning M
9. **end if**
10. **Return:** the updated lexicon *Lex*

By reasoning as in the previous subsection we can easily show that subroutine (51) is equivalent to (52) under the assumption of output-drivenness. Indeed, the two underlying concatenations $\mathbf{Bb}_{v_1, ..., v_n}$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1, ..., v_n}$ constructed by (52) at line 6 provide a ‘summary’ of the set $\Gamma(v_1, ..., v_n)$ of pairs of underlying concatenations constructed by (51).

(52) 1. **Require:** a partial lexicon *Lex*, a (possibly partial) grammar G, a paradigm π
2. **Require:** a pair (Mᴍ, $\widehat{\mathrm{M}}$ᴍ) of meaning concatenations which share, say, the suffix meaning ᴍ
3. **Require:** a feature φ unset by *Lex* for M, $\widehat{\mathrm{M}}$ or ᴍ; for concreteness, say it is unset for M
4. **Require:** some features $\psi_1, \dots, \psi_n$ all unset for the shared suffix meaning ᴍ according to *Lex*
5. read the surface realisations **Xx** and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of Mᴍ and $\widehat{\mathrm{M}}$ᴍ off the paradigm π
6. **For** any combination $v_1, \dots, v_n$ of values of the features $\psi_1, \dots, \psi_n$, construct the two underlying concatenations $\mathbf{Bb}_{v_1,\dots,v_n}$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1,\dots,v_n}$, as follows:
   - $\mathbf{Bb}_{v_1,\dots,v_n}$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1,\dots,v_n}$ are consistent with *Lex* as underlying forms for Mᴍ and $\widehat{\mathrm{M}}$ᴍ
   - **B** has the opposite value for feature φ than **X**
   - $\mathbf{b}_{v_1,\dots,v_n}$ and $\hat{\mathbf{b}}_{v_1,\dots,v_n}$ have features $\psi_1, \dots, \psi_n$ set to the values $v_1, \dots, v_n$
   - $\mathbf{Bb}_{v_1,\dots,v_n}$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1,\dots,v_n}$ are otherwise identical to **Xx** and $\hat{\mathbf{X}}\hat{\mathbf{x}}$
7. **if** the grammar G is inconsistent with the two mappings ($\mathbf{Bb}_{v_1,\dots,v_n}$, **Xx**) and ($\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1,\dots,v_n}$, $\hat{\mathbf{X}}\hat{\mathbf{x}}$) for any combination of values $v_1, \dots, v_n$ **then**
8. update the lexicon *Lex* by setting feature φ to that constant value for the meaning M
9. **end if**
10. **Return:** the updated lexicon *Lex*

If the features $\psi_1, \dots, \psi_n$ considered in line 4 of (52) exhaust the set of features, then the two subroutines (49) and (52) are equivalent. However, the running time of (52) is exponential in the number of features in the latter case, because it needs to consider all combinations $v_1, \dots, v_n$.

4 Subroutines for extracting ranking information from a contrast pair

By reasoning as in the previous subsection, we can proceed from Merchant's original subroutine (41) for extracting ranking information from contrast pairs to the variant in (53).

(53) 1. **Require:** an ERC matrix $\mathfrak{R}$, a (possibly partial) lexicon *Lex*, a paradigm π
2. **Require:** a pair (Mᴍ, $\widehat{\mathrm{M}}$ᴍ) of meaning concatenations which share, say, the suffix meaning ᴍ
3. **Require:** some features $\psi_1, \dots, \psi_n$ all unset for the shared suffix meaning ᴍ according to *Lex*
4. read the surface realisation **Xx** and $\hat{\mathbf{X}}\hat{\mathbf{x}}$ of Mᴍ and $\widehat{\mathrm{M}}$ᴍ off the paradigm π

5. **For** any combination $v_1, \ldots, v_n$ of values of the features $\psi_1, \ldots, \psi_n$, construct the set $\Gamma(v_1, \ldots, v_n)$ of pairs ($\mathbf{Aa}$, $\widehat{\mathbf{A}}\widehat{\mathbf{a}}$) of underlying concatenations such that
   - $\mathbf{Aa}$ is consistent with the lexicon *Lex* as an underlying form for MM
   - $\widehat{\mathbf{A}}\widehat{\mathbf{a}}$ is consistent with the lexicon *Lex* as an underlying form for $\widehat{\text{M}}$M
   - the two underlying suffix morphemes **a** and $\widehat{\mathbf{a}}$ both have the values $v_1, \ldots, v_n$ for the features $\psi_1, \ldots, \psi_n$
6. compute the join of the ERC matrices $\mathfrak{R}(\mathbf{Aa}, \mathbf{Xx}) + \mathfrak{R}(\widehat{\mathbf{A}}\mathbf{a}, \widehat{\mathbf{X}}\widehat{\mathbf{x}})$ for all pairs ($\mathbf{Aa}$, $\widehat{\mathbf{A}}\widehat{\mathbf{a}}$) in $\Gamma(v_1, \ldots, v_n)$ for all combinations of values $v_1, \ldots, v_n$
7. add this join to the ERC matrix $\mathfrak{R}$
8. **Return:** the updated ERC matrix $\mathfrak{R}$

Once again, if we do not consider any ψ features at line 3, (53) effectively considers the two meaning combinations of the contrast pair independently of each other, and thus ends up equivalent to subroutine (40) for extracting lexical information from single meaning combinations. If we instead consider all unset ψ features at line 3, then (53) effectively only considers pairs of underlying forms for the two meaning combinations which share the underlying suffix, and is therefore equivalent to Merchant's original subroutine (41) for extracting lexical information from contrast pairs. Ultimately, depending on the number of ψ features considered at line 3, (53) provides a whole range of subroutines for extracting ranking information in between the weaker (40) and the stronger (41).

By reasoning as in §2 of this appendix, we can easily show subroutine (53) to be equivalent to (54) under the assumption of output-drivenness. The two underlying concatenations $\mathbf{Bb}_{v_1, \ldots, v_n}$ and $\widehat{\mathbf{B}}\widehat{\mathbf{b}}_{v_1, \ldots, v_n}$ constructed by (54) at line 5 again provide a 'summary' of the set $\Gamma(v_1, \ldots, v_n)$ of pairs of underlying concatenations constructed by (53).

(54)
1. **Require:** an ERC matrix $\mathfrak{R}$, a (possibly partial) lexicon *Lex*, a paradigm π
2. **Require:** a pair (MM, $\widehat{\text{M}}$M) of meaning concatenations which share, say, the suffix meaning M
3. **Require:** some features $\psi_1, \ldots, \psi_n$ all unset for the shared suffix meaning M according to *Lex*
4. read the surface realisation $\mathbf{Xx}$ and $\widehat{\mathbf{X}}\widehat{\mathbf{x}}$ of MM and $\widehat{\text{M}}$M off the paradigm π
5. **For** any combination $v_1, \ldots, v_n$ of values of the features $\psi_1, \ldots, \psi_n$, construct the two underlying concatenations $\mathbf{Bb}_{v_1, \ldots, v_n}$ and $\widehat{\mathbf{B}}\widehat{\mathbf{b}}_{v_1, \ldots, v_n}$ as follows
   - $\mathbf{Bb}_{v_1, \ldots, v_n}$ and $\widehat{\mathbf{B}}\widehat{\mathbf{b}}_{v_1, \ldots, v_n}$ are consistent with *Lex* as underlying forms for MM and $\widehat{\text{M}}$M

- $\mathbf{b}_{v_1, \ldots, v_n}$ and $\hat{\mathbf{b}}_{v_1, \ldots, v_n}$ have features $\psi_1, \ldots, \psi_n$ set to the values $v_1, \ldots, v_n$
- $\mathbf{Bb}_{v_1, \ldots, v_n}$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1, \ldots, v_n}$ are otherwise identical to $\mathbf{Xx}$ and $\hat{\mathbf{X}}\hat{\mathbf{x}}$

6. compute the join of the ERC matrices $\mathfrak{R}(\mathbf{Bb}_{v_1, \ldots, v_n}, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{B}}\hat{\mathbf{b}}_{v_1, \ldots, v_n}, \hat{\mathbf{X}}\hat{\mathbf{x}})$ for all combinations of values $v_1, \ldots, v_n$
7. add this join to the ERC matrix $\mathfrak{R}$
8. **Return:** the updated ERC matrix $\mathfrak{R}$

Take the simplest case, where we consider a single feature ψ at line 3. Assume furthermore that ψ is binary, i.e. it only takes the values v = '+' and v = '−'. At line 5, the subroutine constructs the two underlying forms $\mathbf{Bb}_+$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_+$, corresponding to the value v = '+'of feature ψ (i.e. both $\mathbf{b}_+$ and $\hat{\mathbf{b}}_+$ have the value '+' for the feature ψ), and the two underlying forms $\mathbf{Bb}_-$ and $\hat{\mathbf{B}}\hat{\mathbf{b}}_-$, corresponding to the value v = '−' (i.e. both $\mathbf{b}_-$ and $\hat{\mathbf{b}}_-$ have the value '−' for the feature ψ). At line 6, we finally compute the join of the two ERC blocks $\mathfrak{R}(\mathbf{Bb}_+, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{B}}\hat{\mathbf{b}}_+, \hat{\mathbf{X}}\hat{\mathbf{x}})$ and $\mathfrak{R}(\mathbf{Bb}_-, \mathbf{Xx}) + \mathfrak{R}(\hat{\mathbf{B}}\hat{\mathbf{b}}_-, \hat{\mathbf{X}}\hat{\mathbf{x}})$. Even in the simplest case of a single feature ψ, there is no way around computing the join.

In summary, Merchant develops a new subroutine for extracting ranking information through the notion of join. This subroutine can be applied to either a single meaning combination at a time, as in (40); or to a contrast pair, as in (41). The latter is more powerful than the former: the subroutine extracts more ranking information by processing the two meaning combinations Mm and $\widehat{\text{M}}$m as a contrast pair than by processing them separately. The crux of Merchant's subroutine for extracting ranking information is that it relies on the computation of a join, which is time-consuming. The biggest impact of output-drivenness is that it allows Merchant's subroutine for extracting ranking information from a single meaning combination to be restated without having to compute any join. The restated subroutine thus runs in time linear in the number of features. In other words, it is efficient even relative to the more demanding notion of efficiency introduced in §2.6. The situation is unfortunately very different when Merchant's subroutine for extracting ranking information is boosted by being applied to contrast pairs. In this case, output-drivenness is not able to circumvent the computation of the join, and thus does not suffice to make the subroutine efficient.