Learning the Dutch plural in Optimality Theory

Peter Bourgonje
Utrecht University

MA. Thesis Linguistics
Supervised by prof. dr. Wim Zonneveld and prof. dr. Rene Kager

August 2010

Table of contents

Contents

Acknowledgements

First and foremost I would like to thank Wim Zonneveld. Perhaps without realizing it, both in the lessons I took from him and during supervising the writing of my thesis, he managed to arouse my interest in theoretical linguistics in great depth, and encouraged me to come up with new ideas to solve problems at hand. The comments on earlier versions of this thesis were sometimes annoyingly correct and to the point, but were always of such nature that they encouraged me to write a better manuscript and deliver better work. I can truly say that in the last stage of my studies, his suggestions helped me a great deal and I am thankful for having had Wim as a teacher and supervisor.

I would also like to thank Rene Kager for taking the time to be the second reader for this thesis. His comments were to the point, honest and helpful.

Last but not least, I would like to thank Bert Bourgonje, my father, for the many, many endless discussions about linguistic issues, most of them which ended far past midnight, but never too late. Always eager to hear about my progress, and never reluctant in showing enthusiasm and providing helpful insights, he helped me along the way many times. Without all this, I probably never would have ended up becoming or being a linguist in the first place.

Abstract

This thesis deals with language acquisition within the framework of stochastic Optimality Theory. Stochastic Optimality Theory was specifically designed to model variation and optionality, which traditionally are critical areas for standard Optimality Theory.
Using a learning algorithm that comes with stochastic OT – the Gradual Learning Algorithm proposed by Boersma & Hayes (2001) – I will try to learn the variation in Dutch noun pluralisation. The data set is taken from Van Wijk (2007) and contains the distribution of the two default plural affixes /-en/ and /-s/ over various phonological contexts. In some phonological contexts, Dutch noun pluralisation displays variation.

This acquisition experiment aims to test the Gradual Learning Algorithm as a plausible device for language acquisition. If the algorithm succeeds in learning the correct grammar – or constraint hierarchy in Optimality Theoretic terms – it will prove to be yet a more interesting approach to consider in language acquisition research.
Results show however that the Gradual Learning Algorithm has severe difficulties in constructing a grammar that correctly models the variation in Dutch noun pluralisation.

1. Introduction

The general topic of this thesis is language acquisition in an Optimality Theoretic framework.
More specifically, I will use a learning algorithm to learn the variation that exists in the Dutch pluralisation of nouns.
The productive affixes to form the plural of nouns in Dutch are /-s/ and /-en/. Which affix adheres to which noun is determined by the phonological context. In some contexts however, variation appears; both affixes can appear after the same noun. An example of this is the noun '*hersen*', the plural of which can be either '*hersens'* or '*hersenen*'.

To learn the variation in Dutch pluralisation within the framework of Optimality Theory (OT), I will use a set of data concerning the Dutch plural. This data is based on empirical research done by Van Wijk (2007). In an attempt to describe the status of the default plural affix in Dutch, Van Wijk asked several groups of subjects to produce the plural form of non-existent words. I will use the data she obtained to get an overview of the distribution of the plural affix in the relevant and specified phonological contexts.

In the learning task I have chosen for an Optimality Theoretic learning algorithm. In comparison to the preceding (generative and rule-based) frameworks of language and language acquisition, OT has proven to be particularly successful in the area of phonology. The idea of violable constraints and the locus on output – instead of input and/or derivations from the input – has brought along many new insights and improvements in doing phonology.
Deriving from this theory of standard OT (Prince & Smolensky, 1993) is the theory of stochastic OT, and the learning algorithm that comes with it (Boersma & Hayes, 2001). This 'version' of OT was particularly intended to handle variation and optionality, which are traditional problems for standard OT.

The aim of this thesis is to test this learning algorithm for stochastic OT, the Gradual Learning Algorithm, on the data of the Dutch pluralisation of nouns.
I will provide the Gradual Learning Algorithm with the constraints it needs to build up a grammar, then feed the data concerning Dutch pluralisation – taken from Van Wijk – and test if the learning algorithm can come up with the correct grammar to produce the plural of Dutch nouns correctly.

If the algorithm proves to be able to learn the variation in Dutch pluralisation, by reproducing the correct distributions over the phonological contexts, this is another argument to consider stochastic OT - and in particular the Gradual Learning Algorithm – as a plausible theory of language and language acquisition.

This thesis is divided into two parts. In the first part, in chapter 2, Optimality Theory will be discussed. In chapter 3, I will discuss language acquisition in the framework of OT. Chapter 4 provides an overview of optionality and variation in language; based on explanations of particular cases from Antilla and Hayes.
In chapter 5 the distribution of the Dutch plural is discussed. Chapter 6 and 7 present two alternative approaches to variation in the OT framework, respectively Antilla's and Hayes'.
Chapter 8 introduces the Gradual Learning Algorithm; its design and advantages.

The second part of this thesis describes the learning experiment and the results.
In chapter 9, the set-up of the experiment is described and the results are presented.
Chapter 10 contains the evaluation and discussion of the results and provides some suggestions for further research. Finally, chapter 11 briefly summarizes the major conclusions of this thesis.

2. Introduction to Optimality Theory

Optimality Theory (OT) was introduced as a theory of language phenomena in the early 90's of the previous century. It proposed the - at least in linguistics - new idea of approaching language material exclusively by constraints, to the detriment of rules, parameters and parameter settings, and similar devices. In fact, it is no exaggeration to say that OT arose out of a dissatisfaction with how its major predecessors, especially the rule-based framework of the 1960s and 1970s and Principles and Parameters theory of the 1980s, handled language material and language phenomena.
Although these theories recognized - up to a certain degree - the role of constraints, the step taken by OT was to allow only constraints in its analyses, and interestingly, just constraints that could justifiably be called 'universal'. The rapidly growing popularity of OT in the 1990s and the first decade of the current century showed that this was seen by many as a major step forward in doing linguistics. How does OT implement these steps and how does it live up to its aims?

As McCarthy & Prince (1994) put it, OT has five basic principles.
The first is **universality**. The set of constraints that OT works with are provided by Universal grammar (UG). These constraints are universal and present in every particular, language-specific grammar.
The second is **violability.** Essential in OT is that the constraints can be violated. If a constraint is violated, it does not mean that the output violating this constraints is immediately out.
The third assumptions is that the constraints, provided by UG, have a certain **ranking**. This ranking is language-specific (i.e. differs from language to language). In OT, a *grammar* is the equivalent of a hierarchy of the constraints.
The fourth assumption is **inclusiveness**. The constraint hierarchy "evaluates a set of candidate analyses that are admitted by very general considerations of structural well-formedness".
The fifth and final assumption is **parallelism**. The optimal candidate – the candidate that minimally violates the constraint hierarchy – is computed over the complete hierarchy and candidate set and there is no serial derivation.

UG must minimally provide the following:
As noted, UG provides the set of constraints. In OT introductions typically referred to as **con**.
Then, UG provides the function **gen**. For each input (*i*), *gen* defines a range of candidates available to this input (*i*). So to say, the output candidate the speaker – given *i* – can 'choose from'.
Finally, UG provides **eval**. *eval* is a function evaluating the forms or output candidates (generated by gen) with respect to the constraint hierarchy – the ranking.

When the function *gen* generates a list of candidate analyses for an input *i*, the function *eval* rates the members of the list. Given the ranking of *con*, *eval* determines the relative harmony of the candidates on the list. The *maximally harmonic* candidate, which also *minimally violates* the constraints given their ranking, is the optimal candidate. This will become the output associated with *i*.

Typically, a *tableau* like the one below is used to illustrate the evaluation process.

Candidates	A	B
>> **k-cand$_1$**		*
k-cand$_2$	!*	

Tableau 1: Evaluation of candidates in OT (*from McCarthy & Prince, 1993a, p. 87*)

The constraints in OT are of two different types. On the one hand there are markedness constraints. The purpose of these constraints is to rule out certain 'marked', or complicated structures. (For a discussion of what 'marked' means, see McCarthy & Prince, 1994.)
Markedness constraints evaluate candidates, using markedness as a criterion. It depends on the hierarchy of the grammar (e.g. whether or not the relevant markedness constraints are high-ranked) if this marked structure becomes the optimal output and surfaces or not.

The other type of constraints are the faithfulness constraints. The task of the faithfulness constraints is to ensure that the output does not differ from the input.
What follows is a brief illustration of how constraint interaction and the selection of the optimal candidate works in OT.

Given a small set of constraints:
- Obstruent clusters must be homogeneous with regard to the feature voice (VOICE-HOM)
- Avoid adjacent coronal plosives (*CORCOR)
- Avoid insertion (*INS)

The first two constraints are markedness constraints (prohibiting some marked structure), the third is a faithfulness constraint (demanding 'faithfulness' to the input form).

With the past tense form of the English verb *to want* as an input, the following tableau emerges:

want+past tense	VOICE-HOM	*CORCOR	*INS
want-t		!*	
want-d	!*	*	
>> **want-ed**			*

Tableau 2: OT analysis of the past tense form of the verb *to want*

Starting with the second candidate (want-d), this candidate violates the VOICE-HOM constraint. This is the highest ranked constraint of the three. The exclamation mark indicates that this violation rejects the second candidate as the optimal candidate. In fact, the violation of the constraint *CORCOR is not even relevant anymore. The first candidate (want-t) does not violate the top-ranked VOICE-HOM constraint. But it does violate the *CORCOR constraint. Again, the exclamation mark shows that this violation rejects want-t as the optimal candidate, since there is another candidate that does not violate the *CORCOR constraint. The only candidate left, which is now optimal, is want-ed. The grey area in the column for the *INS constraint indicates that the violation for this constraint (which the want-ed candidate incurs) is not relevant anymore, since at this point want-ed is the only candidate left.

This example shows the hierarchy (for these three constraints) English needs, to arrive at the data that is observed[1].

It also shows that the two types of constraints are constantly in conflict.
The optimal candidate violates the faithfulness constraint, but satisfies the markedness constraints.
The other two candidates satisfy the faithfulness constraint, but violate the markedness constraints.
The hierarchy determines which candidate comes out as optimal.

[1] In fact it only shows that the two markedness constraints should outrank (be higher ranked than) the faithfulness constraint. The hierarchy that follows from this example is (VOICE-HOM , *CORCOR) >> *ins, and not VOICE-HOM >> *CORCOR >> *INS. However, with other examples, such as the past tense of *to help* and *to stab*, it can be shown that this is actually the ranking needed to arrive at the correct grammar for the English language.

3. Optimality Theory and acquisition

The topic of this thesis being language acquisition in the framework of OT, I will now describe how language acquisition in the OT framework proceeds.

If grammars in OT are represented by constraint hierarchies, and different languages have different grammars, then different languages are represented by different constraint hierarchies. The next example, based on the phonological process of final devoicing, shows that this is indeed the case.

It is generally known that English does not do final devoicing. It is perfectly acceptable to produce forms like /bed/, /rQb/ and /bVg/. Dutch however, does devoice final obstruents. The underlying form /bEd/ becomes /bEt/, /Eb/ becomes /Ep/.

The constraints regulating this phenomenon could be the following[2]:

- Be faithful to the identity of the onset (with regard to voice features) (IDONSLAR)
- Be faithful to the identity of the input (with regard to voice features) (IDLAR)
- Avoid voiced obstruents (*LAR)

Note that the constraint IDONSLAR only applies to onsets, IDLAR only applies to codas.

If we assume the following hierarchy with regard to this set of constraints:
IDONSLAR >> IDLAR >> *LAR
We arrive at the English grammar[3], as shown in tableau 3.

/bed/	IDONSLAR	IDLAR	*LAR
>> **/bed/**			**
/bet/		!*	*
/ped/	!*		*
/pet/	!*	*	

Tableau 3: OT analysis for the underlying form /bed/ in English

The optimal form /bed/ incurs two violations of the constraint *LAR, because it has two voiced obstruents. The output form /bet/ incurs one violation of the constraint *LAR, because of the single voiced obstruent. And one violation of the constraint IDLAR, because the final obstruent in the input has become voiceless. The output form /ped/ violates IDONSLAR, because the obstruent in the onset has become voiceless, and violates *LAR because of the voiced obstruent in the coda.

2 These constraints are frequently used in the relevant literature.
3 As in figure 2, this data alone does not suffice to prove this hierarchy. But other input/output pairs are available to prove that this is in fact the correct hierarchy for English.

Finally, the output form /pet/ violates both IDOnsLar and IDLar, because both obstruents have become voiceless (and as a result it does not violate *Lar at all).

The Dutch analysis of the same underlying form /bEd/[4] can be done with the same set of constraints (and in fact must be done with the same set; the universality principle of OT). But with a different ranking.
If we assume the following ranking:
IDOnsLar >> *Lar >> IDLar
we arrive at the Dutch tableau (this time without the output forms with alternative onsets), as shown in tableau 4.

/bEd/	IDOnsLar	*Lar	IDLar
/bEd/		*!*	
>> /bEt/		*	*

Tableau 4: OT analysis for the underlying form /bEd/ in Dutch

The output form /bEd/ incurs two violations on the *Lar constraint, because of the voiced obstruents in the onset and coda. The output form /bEt/ violates *Lar, because of the voiced obstruent in the onset, and violates IDLar because the obstruent in the coda has become voiceless.
The second violation of the *Lar constraint for the output form /bEd/ results in the output form /bEt/ being optimal. As the grey area shows, the fact that /bEt/ does violate IDLar and /bEd/ does not, is not relevant.

As the example of final devoicing shows, with ranking *x* we arrive at the English grammar, whereas with the same set of constraints, but ranking *y* we arrive at the Dutch grammar.

In the language acquisition task, when arriving at a certain ranking, we arrive at a certain language. To learn a language, now ‘only’ means to rearrange the constraints into a certain ranking. If we put some set of constraints in the order A, we might end up with Swahili for instance, and if we put this same set of constraints in the order B, we might end up with Japanese. It is important that every single constraint is present in every natural language and every natural language has every single constraint.

The task of language acquisition is now defined as ‘simply’ putting the constraints (which are innate and already available through UG) in the right order, and one will end up with the right language. Along the way, the language learner obviously encounters language data that does or does not conform to its current constraint ranking. If it does not conform to the current ranking, the learner should re-adjust the ranking, so that this new ranking can also produce this new data correctly.

[4] The difference in spelling (/bEd/ in Dutch, /bed/ in English for the underlying forms) stems from the fact that SAMPA keeps a language specific scheme for phonetic notation.

But before discussing the way in which learning proceeds, the right question to ask at this point is: What is the ranking the language learner starts off with?

As it turns out, the initial ranking that has to be assumed for OT is not an arbitrary one. A more general procedure of acquisition (not necessarily language acquisition) was already introduced in the Principles & Parameters framework. This procedure is called the **subset principle** (Baker, 1979). If language acquisition is taken to be data-driven, meaning that a learner encounters data and constantly has to re-adjust his hypothesis about how the system works, particularly when data is encountered that does not conform to the current hypothesis. Then it follows that it is impossible to get from the larger system to the smaller system. This is because the larger system always will include the data from the smaller system. This, of course, is related to the well-known phenomenon (in the language acquisition field) of the **negative evidence** argument.

For example, imagine language A (like English), which allows both voiceless and voiced final obstruents. And imagine language B (like Dutch), which allows only voiceless final obstruents. If the grammar of language A was the starting point, it is impossible to get to the grammar of language B, because any data that the learner will encounter (e.g. voiceless final obstruents) will always also conform to the grammar of language A.
Generally, since the larger language includes the smaller language, there will never be any evidence that the grammar of the larger language does not suffice. And just from the fact that something does never emerge, it cannot be concluded that it therefore cannot emerge at all.

From this principle, it follows that the initial ranking in OT must be that the markedness constraints are ranked higher than the faithfulness constraints. With this initial ranking, everything that is marked, does not conform to the current grammar (which is the smallest grammar imaginable). The learner will at some point undoubtedly be confronted with some marked structure. This means re-adjusting the ranking, to arrive at the correct grammar. This is equivalent to learning via the subset principle. Thus it follows that initially, markedness constraints outrank faithfulness constraints.
Once this is the initial point, 'all the learner has to do' is demoting the markedness constraints to a particular point, and/or promoting the faithfulness constraints to a particular point to arrive at the correct ranking for the particular grammar.

This initial ranking, which is logically necessary, is also funded with empirical proof. For a discussion on empirical proofs that show markedness must initially outrank faithfulness, see McCarthy & Prince, 1994.

Before moving on to a discussion of optionality and variation in language (which is the phenomenon for which I will try to learn a correct grammar in this thesis), I would like to make two final remarks within the topic of OT and acquisition in OT.

As explained in chapter 2, part of the language device in the framework is the function **gen**. This function generates output candidates (to appear in the tableaus). The reason I have not discussed this function in further detail, particularly with respect to what forms it does and what forms it does not generate, is because there is actually no restriction on (nature of the) output candidates it generates. The list, simply, is assumed to be infinite. For each input *i,* there is a infinite amount of output candidates to choose from. This is simple and elegant from a theoretical point of view, because there is no complicated specification needed for this generator function.
However, it clearly is complicated and in its strictest sense in fact impossible from a psychological point of view. The infinity that is introduced, should directly disqualify OT as a plausible theory of language and language acquisition. Given our finite brain, infinities are obviously problematic.

But as we proceed trying to model natural language in OT and come up with learning algorithms to learn natural language in the OT framework, and if we succeed in that to an increasing degree. Then we must at some point come up with a way around this infinity.
One at least formally possible and plausible way to do this, would be with the help of a (limited) Levenshtein distance (Levenshtein, 1966). Given a certain input, the generator function must generate every output that deviates up to *n* units from the input. The units in the phonological area could very well be phonemes. If *n* is chosen correctly, it will include all relevant and/or interesting candidates, but the number of output candidates will not be infinite.
This problem, however interesting, is beyond the scope of this thesis. As a more theoretical survey, I will only consider the relevant (e.g. those relevant for the particular constraints) output candidates given each input, as usually done in OT analysis.

The second remark I would like to make, before moving on, concerns the **con** set of chapter one. Con holds all constraints, all of which available through UG. It states that all constraints are language universal and present in all languages (or grammars).
However, it does not tell us which constraints these are, and how they can be defined. Clearly, if constraints 'come for free' because they are universal by definition, one might be tempted to simply invent new constraints on the go, to handle specific cases in specific languages. These new constraint can then just be low-ranked in all other languages, rendering them inactive in these languages.

To prevent this proliferation of constraints, there is a constraint on the invention of new constraints. This supra-constraint follows from a phenomenon called **factorial typology**. Factorial typology means that every hierarchy given the set of constraints should be represented by an actual natural language. If one invents some constraint C, that should be in position *k* to arrive at the German grammar, putting this constraint C in position *k+3* (with two constraints in between *k* and its new position), should result in another actual natural language.

Even though not every single hierarchy (given the nature of the constraints) will produce another language (so the ‘price to pay’ for adding a new constraint does not result in an exponential number of new possible languages), adding a new constraint to the set is not something that should be done without any further consideration. This of course protects the ‘universalness’ of the set.

With these remarks made, I will now move on to the discussion of optionality and variation in language.

4. Optionality and variation

If the acquisition task is defined as arriving at the correct ranking, and the ranking is supposed to be constant and strict (for another ranking is associated with another language), then it is clear that optionality and free variation in language is problematic for OT. Given a certain hierarchy and strict ranking, some input always has the same optimal output. No variation can occur. If we want another candidate to be optimal, we have to change the ranking of the relevant constraints. But this would theoretically result in (evaluating under) another language. This is not what we assume when free variation or optionality is meant.

Optionality and variation as OT problems were recognized in the course of the first decade of OT's existence. Antilla (1997) was one of the first works to do so: "*Variation is a notorious problem for generative theories of language structure. Especially problematic is the fact that variation is hardly ever totally free: the environment does not determine the variant categorically, yet it gives rise to a statistical bias.*". In this work, he exemplifies the intricate case of the genitive plural in Finnish, trying to make sense of the optional variation between suffix forms in certain contexts. Because the topical case of this thesis schematically implies a similar situation, it would be helpful to have a look at Antilla's discussion.

In this chapter, I will discuss two cases of (broadly speaking) 'optionality'. One is the case of the genitive plural in Finnish actually discussed by Antilla (1997), the other is the case of light vs. dark /l/ discussed by Boersma and Hayes (2001). The purpose of this discussion is to explain the 'context-senstivity' that is usually characteristic to optionality or variation. Optionality and variation are not just found everywhere, but usually only in very specific contexts or very specific cases. Furthermore, as the above quote from Antilla states, the environment typically has some statistical bias towards one option or the other.

The two suffixes available for the genitive plural in Finnish can be divided into strong and weak. Strong is /*-iden*/ and weak is either /*-en*/ or /*-jen*/. Monosyllabic stems unexceptionally take the strong variant (see table 1: A). For disyllabic stems, the choice for either strong of weak depends on the syllable structure of the final syllable. Light final syllables take the weak variant (table 1: B), others take the strong variant (table 1: C).

A	*/maa/*	*'land'*	*mai.den*
	/tie/	*'road'*	*tei.den*
B	*/kala/*	*'fish'*	*ka.lo.jen*
	/lasi/	*'glass'*	*la.si.en*
C	*/palttoo/*	*'coat'*	*palt.toi.den*
	/varas/	*'thief'*	*var.kai.den*

Table 1: Genitive plural suffix in Finnish in various contexts *(from Antilla, 1997, p.38)*

Variation enters the picture with trisyllabic stems with a light final syllable. Stems ending in a high vowel preferably take the weak variant (table 2: D), stems ending in a low vowel preferably take the strong variant (table 2: E) and the mid vowels do not show a clear preference (table 2: F).

D	*/lemmikki/*	*'pet'*	*lem.mik.ki.en* ~ (*lem.mi.kei.den*)
	/sihteeri/	*'secretary'*	*sih.tee.ri.en* ~ (*sih.tee.rei.den*)
E	*/korjaamo/*	*'repair shop'*	*kor.jaa.mo.jen* ~ (*kor.jaa.moi.den*)
	/fyysikko/	*'physicist'*	*fyy.sik.ki.jen ~ fyy.si.koi.den*
F	*/sairaala/*	*'hospital'*	*sai.raa.loi.den ~ sai.raa.lo.jen*
	/kamera/	*'camera'*	*ka.me.roi.den* ~ (?*ka.me.ro.jen*)

Table 2: Genitive plural suffix in Finnish in various contexts *(from Antilla, 1997, p.40)*

The non-preferred, but crucially not ungrammatical variants are shown between brackets.
This is a clear case of variation in grammar. In Chapter 6 I will return to this case, briefly describing the way Antilla actually models this type of variation.

A phenomenon closely related to variation, but not exactly the same, is that of gradient well-formedness. Slightly different from variation, a case of gradient well-formedness occurs when subjects point out that something is neither ungrammatical nor perfectly well-formed. It is 'somewhere in between', and subjects usually respond to cases like these as "*I would not do it that way myself, but I guess it's okay*". Boersma & Hayes (2001) is a well-known OT paper tackling this issue, one of their cases being the choice between dark and light */l/* in English, in certain contexts. This is their description:

"*In various American English dialects, /l/ is obligatory light in two positions: initial (*Louanne, Light*) and pretonic (*allow*, and again* Light*). It is obligatory dark in final and preconsontantal position (*bell*,* help*). In medial, pre-atonic position there is free variation: forms like* Greeley *can have light or dark /l/.* " (Boersma & Hayes 2001, p. 28)

Furthermore, there are some effects of morphology. For /l/-final stems with a vowel-initial stressless suffix, there is a strong preference for the dark /l/. This preference is even stronger for similar cases, but with a word break (e.g. *mail it* instead of *feely*). The other way round: for vowel-final stems with /l/-initial suffix, there is a preference for a light /l/.

Boersma & Hayes propose a set of constraints, based on the above mentioned facts, to model the distribution below, which is based on the judgments of ten native speakers elicited a purposely devised test (with a 1 for typically light and a 7 for typically dark).

		Mean rating as light	*Mean rating as dark*	*Judgment difference*
1	*Light*	*1.30*	*6.10*	*4.80*
2	*Louanne*	*1.10*	*5.55*	*4.45*
3	*gray-ling, gai-ly, free-ly*	*1.57*	*3.34*	*1.77*
4	*Mailer, Hayley, Greeley*	*1.90*	*2.64*	*0.74*
5	*mail-er, hail-y, feel-y*	*3.01*	*2.01*	*-1.00*
6	*mail it*	*4.40*	*1.10*	*-3.30*
7	*bell, help*	*6.60*	*1.12*	*-5.48*

Table 3: Distribution of dark and light /l/ in various contexts *(from Boersma & Hayes, 2001, p. 33)*

As table 3 shows, groups 1 and 7 have a near-categorical distribution, with either dark or light /l/ completely ungrammatical. Groups 3, 4 and 5 show a much more diffused distribution. Just as in the Finnish case, I will return to the actual modeling of this case by the authors in Chapter X.

As explained in the introduction, the actual topic of this thesis is Duch pluralisation of nouns. Everything presented so far in this thesis can be seen as background to that discussion. Before turning to that topic, however, it is important to make one final note on (linguistic) variation in general.

When looking at optionality and variation, it rapidly springs to mind that the line between a language or dialect on the one hand and true variation on the other, can actually be quite thin. Consider, for instance, the following case of Dutch word order.

1. *Ik vroeg of je hem gezien hebt.*
   I asked if you him seen have
   "I asked if you've seen him."
2. *Ik vroeg of je hem hebt gezien.*
   I asked if you him have seen
   "I asked if you've seen him."

From the point of view of Dutch grammar, this seems to be a case of optionality, both orders simply counting as 'Dutch': these sentences mean exactly the same and can be used in exactly the same contexts. However, a closer, sociolinguistic look at this phenomenon reveals that 1 is the so-called 'green order', which is associated with and most often used in the eastern part of the Netherlands, whereas 2, the so-called 'red order', is most often used in the other parts of the Netherlands. Thus, this seeming example of optionality probably has a more accurate interpretation as one of different dialects, and thus probably of different grammars for different speakers. These dialects can be called dialects of the same language because no mutual intelligibility problems arise for these speakers, at least not ones involving this word order phenomenon.

Regarding terminology such as language and dialect and issues surrounding them, the strictest view of a grammar, thus of variation in a grammar (and one I tend to adhere to) is that of a system that only exists in individuals. This is what Chomsky formulates as “I-Language”: “*The only (virtually) “shared structure” among humans generally is the initial state of the language faculty. Beyond that we expect to find no more than approximations, as in the case of natural objects that grow and develop.”* (from *New horizons in the study of language and mind*, Chomsky, 2000, chapter 2 p.30) Chomsky then proceeds to claim that the internal representation of language inside the minds of speaker A and B (who speak language *x)* are merely alike (and alike more than the internal representation of language inside the mind of speaker C, who (only) speaks language *y*), but are essentially not the same.

According to this view, a grammar is something that only exists within the head (brain) of a single individual, and not something that exists within a community. Variation in a grammar should therefore also exist within the head (brain) of a single individual, implying that one speaker, when exhibiting variation in his speech, should actually utter both options on different occasions, i.e. more or less alternately. Against these remarks, it is worth noting that the data collected by Van Wijk, which will be the empirical material underlying the discussion in the next chapter, do show speaker internal variation, with one subject usually choosing candidate A, but occasionally choosing alternative candidate B, truly ‘optionally’. So even according to the narrowest view of grammar and variation in grammar, as something that really only exists within a single speaker’s head, the data I will address will turn out to show ‘optionality’/’variation’.

5. The Dutch Plural

The data that will be the empirical basis for this topic is collected by Van Wijk (2007).
First, I will give a brief introduction to the aims to which she collected this data on the Dutch pluralisation of nouns.

Van Wijk's dissertation is about language acquisition, specifically investigating the status of the Dutch defaults forms for noun pluralisation.
This is done against a central theory concerning language acquisition; Pinker's *Dual Route* Model and the '*Words-and-Rules'* theory (Pinker & Prince, 1991). This theory mainly deals with inflectional morphology and the basic notions are that irregular forms (of nouns, but also verbs, adjectives, etc.) are stored in the mental lexicons, and that regular forms are the result of some rule adding a default affix to the stem form[5].

These central claims (concerning regular and irregular forms) are made mainly on the basis of research done on English and German. The data for these languages suggests a single default form for the pluralisation of nouns. In English for example, all regular nouns take the affix /-s/ to form the plural[6].

Dutch however, seems to have two default affixes for noun pluralisation. Both are productive (i.e. can be used on new or unknown words). If the *Words-and-Rules* theory, which assumes a single default affix, is correct, then children during their acquisition phase should look for the one single default in the data they are presented with. The core of the dissertation of Van Wijk is to confront the Dual Route Model with the empirical data from the acquisition of the Dutch plural.

To examine this claim (of a single default), Van Wijk used the Van Dale corpus for research on frequency data. The Van Dale corpus contains a list of 17.298 nouns with their plural form(s). From this, the number of occurrences of all the affixes of interest to this study can be taken. (For a detailed discussion on the Van Dale corpus, see Van Wijk, 2007, p. 44/45.)

First the data set is discussed and Van Wijk explains which principles seem to govern the distribution of the two default affixes (a detailed discussion of this will follow in this chapter).
She then proceeds with testing children during their process of language acquisition on which affix they use for the relevant contexts. The default affix surfaces in the case of speech errors (when speakers utter *mouses* instead of *mice*), but a much easier way to elicit the default is through the commonly used so-called *Wug test*. In a *wug test*, a subject is presented with a picture of some (imaginary) object or creature and its name (in singular form); "*This is a …"*. Then two of those objects or creatures are presented, and the speaker is asked to complete the following sentence: "*There is another …, now there are two…"*.

[5] When the desired form cannot be found in the mental lexicon. The theory does not deny the possibility of also storing (highly frequent) words and their inflection pattern in the mental lexicon.
[6] Although it is obviously not always realized as /-s/, but this is assumed to be because of the phonological context and not because there is more than one default. Consider the plural *cats* as opposed to *dogs* (voiced fricative) and *busses* (voiced fricative and inserted vowel).

Since the names of the objects or creatures are non-existent words, they cannot be stored in the mental lexicon. Therefore, the speaker is forced to use the default affix.

The data Van Wijk based her conclusions on is taken from this production experiment. She used an adult group (of eleven subjects) as control group. For the acquisition phases, seventeen 4-year-olds and eighteen 5-year-olds were taken. (And for a longitudinal experiment, twenty children were followed from 2;8 till 3;3 years old.)
Several non-existent words were carefully chosen representing the various (phonological) contexts of interest, to test speakers' choices for plural suffix.

The results of both the cross-sectional and longitudinal tests show that the theory of a single unique default is not backed up by empirical evidence from the language acquisition process with Dutch children. This leads to the adjustment of the *Words-and-Rules* theory into a *Words-and-Constraints* theory. This in turn forms the basis of the constraints I will use to learn the data set given a specific learning algorithm for OT. All of this will be explained in further detail below.

Having sketched Van Wijk's interests, I will now turn to my own, namely constructing a grammar that can model the variation in the choice for plural affix in certain contexts.

As a first step, consider a brief overview of the Dutch plural affixes in nouns.

	Stem	**Translation**	**Plural**
-en	*boek*	*'book'*	*boek-en*
	kast	*'cupboard'*	*kast-en*
	regenton	*'rain barrel'*	*regenton-(n)en*
	wijnfles	*'wine bottle'*	*wijnfles-(s)en*
	getuigenis	*'testimony'*	*getuigenis-(s)en*
	steekpartij	*'stabbing'*	*steekpartij-en*
	vrouw	*'woman'*	*vrouw-en*
	angst	*'fear'*	*angst-en*
	rivierkreeft	*'crayfish'*	*rivierkreeft-en*
	zitbank	*'couch'*	*zitbank-en*
-s	*versterker*	*'amplifier'*	*versterker-s*
	balkon	*'balcony'*	*balkon-s*
	sofa	*'sofa'*	*sofa'-s*
	formatie	*'formation'*	*formatie-s*
	appel	*'apple'*	*appel-s*
	bioscoopfilm	*'feature film'*	*bioscoopfilm-s*
	oven	*'oven'*	*oven-s*
	tv	*'tv'*	*tv'-s*
	ruimte	*'space'*	*ruimte-s*
	veter	*'shoe lace'*	*veter-s*
-eren	*Kind*	*'child'*	*kind-eren*
	ei	*'egg'*	*ei-eren*
-es	*douche*	*'shower'*	*douch-es*
	lunch	*'lunch'*	*lunch-es*
-i	*politicus*	*'politician'*	*politic-i*
	saldo	*'balance'*	*sald-i*
-a	*museum*	*'museum'*	*muse-a*
	curiosum	*'curiosity'*	*curios-a*

Table 4: Overview of the Dutch plural contexts (note that in some cases, an additional consonant is added ("wijnflessen" for example); this is a spelling convention and does not affect the pronunciation)

All affixes in table 4 from /–eren/ onwards have a limited distribution, co-occurring with idiosyncrasies. The suffix /–eren/ occurs after 30 or so stems; /-es/ only forms the plural of a small group of loan nouns that end in a sibilant; / –i/ and /–a/ are usually associated with nouns of Latin origin, /-i/ being the plural suffix of nouns ending in /–us/ or /–o/ (but far from all of them), and /–a/ the plural of nouns ending in /–um/.

/–en[7]/ and /–s/ are the only really productive affixes for the plural in Dutch. They are by far the most frequent, occur after recent loan nouns (*'sudoku's', 'inbox-en'*), and after language internally formed

[7] Even though the vowel of this suffix is usually reduced to a *schwa*(@) and the consonant is usually deleted in actual pronunciation, I will stick to this /-en/ notation. Which is, then, obviously not a SAMPA notation. The same holds for the other suffixes; /eren/, /-es/, /i/ and /a/. These are also not in SAMPA notation.

new formations (*HAVO's, metroseksuelen*). To a large extent (and this extent is one of the topics of Van Wijk's dissertation) which suffix goes with which noun is determined by the phonological context. Although the rules to regulate this are more like tendencies[8], some predictions regarding the used affix can be made on the basis of the phonological context after which the plural suffix is inserted.

The distribution of the two regular Dutch plural suffixes is determined, or better: influenced, by the two phonological factors of Stress and Sonority. Sometimes these two factors clash and the interaction of these factors causes – what Van Wijk calls - grey areas. These are, predictably, the areas in which variation occurs and will be the focus of this study.

[8] Consider the pairs *'kok-koks' (chef – chefs)* and '*klok – klokken' (clock – clocks)*

5.1 The sonority factor

In phonology, Sonority is considered an inherent property of speech sounds, and moreover, a property that can be expressed using a scale. One common version of the scale is shown below:

Vowels –	*Sonorant Consonants (Liquids, Nasals) –*	*Obstruent consonants*
High Sonority		*Low Sonority*

Finer distinctions are possible, but this three-way scale can be called uncontroversial (Selkirk, 1983). Van Wijk's basic claim involving the sonority factor in Dutch pluralisation is this one:

"As sonority of the final sound [of the noun stem] increases, the likelihood of an /–s/ plural also increases." (Van Wijk, 2007: p. 37).

That is: the strongest preferences for one or the other suffix from the /–en/ - /–s/ pair are at the extremes of the sonority scale.
Nouns ending in back vowels 'practically without exception' take /–s/ as their plural affix (one 'finer' distinction among the vowels being that back vowels are more sonorous than front ones), whereas nouns ending in an obstruent typically take /–en/.
In Van Wijk's view, an explanation of this latter finding, concerning obstruents preferably in combination with /–en/, will run as follows. As is a well-known (maybe even universal) tendency of syllable structure to prefer, in a single syllable, a sonority decrease from the inside out: vowels are typically in the core of the syllable, consonants surround them, and when consonant clusters are found, they – indeed – have the more sonorous consonant closer to the vowel. Dutch words such as *'kracht'* and '*spalk*' are examples of such good syllables. Given this tendency, and relating it to Dutch pluralisation, avoiding /–s/ after obstruent-final stems contributes to avoiding word-final consonant clusters with a sonority plateau or worse: from the syllable sonority perspective, '*boek-en'* is obviously better than '*boek-s'*, because the latter introduces a rise in sonority in the coda.
Notice that, at the other end of the sonority scale, (back) vowels being preferably associated with /–s/, does not immediately follow from syllable sonority considerations; syllable sonority wise '*visie*-s' seems as good as '*visie-en'*. Although sonority clearly plays a role here, this part of its role seems to be less universal.

Two types of Dutch phonemes do not conform to these sonority tendencies. Both glides - /j/, /w/ - (the former not really occurring in coda's) and the three Dutch diphthong vowels -/Ei/, /Au/, /9y/ - which as sound classes are both high on the sonority scale, typically take /–en/ as their plural suffix: '*vrouw-en', 'brouwerij-en', 'schuifpui-en'* and *'bereklau(w)-en'.*

For nouns ending in a sonorant consonant or a front vowel (which are somewhere in the middle of the sonority hierarchy or close to it), either /–s/ or /–en/ can be the plural affix. These are typically the cases where variation appears: *'hersen-s'* vs. *'hersen-en', 'seconde-s'* vs. *'second-en'.*

5.2 The Rhythmic factor

The rhythmic factor of Dutch pluralisation states that:
Nouns should end in a trochaic stress pattern.

Typically, nouns ending in a stressed final syllable (including monosyllabic nouns) prefer /–en/ as the suffix, to create a trochaic stress pattern at the word edge. Nouns with an unstressed final syllable prefer /–s/, either to maintain a trochaic pattern if already present or not to create lengthy sequences of unstressed syllables. Since Dutch can have antepenultimate stress, nouns bearing this stress pattern again show variation in plural affix; /-en/ or /–s/ can be taken. Compare the following example:

1. *kanon – kanonnen (with stress on the second syllable)*
2. *kanon – kanons (with stress on the first syllable)*

The two factors (stress and sonority) determining the choice for plural affix can be conflicting. In fact, they do not agree very often on their individual 'best choice' for plural affix.

	Stressed final syllable	Unstressed final syllable
Diphthong/glide	/-en/	
Obstruent	/-en/	
Sonorant		
Front vowel		
Back vowel		/-s/

Table 5: predictions of plural affix distribution where the two factors agree
(from Van Wijk, 2007, p. 39)

Table 5 represents the contexts discussed above. In the first column all phoneme contexts are listed. In the first row, the stress patterns concerning the final syllable are listed (either stressed or unstressed). If the intersection is empty, this means that the two factors stress and sonority have the opposite preference (the one preferring /–s/, the other /–en/).
If the intersection is not empty, the two factors both have the same preference.
As can be seen in table 5, stress and sonority only agree in the case of a stem with final stress and a diphthong, glide or obstruent as its final phoneme. And in the case of a stem with unstressed final syllable and a back vowel as its final phoneme. In all other cases, the two factors are in conflict.

As can expected, the areas where the two factors agree on the choice for their 'best affix' show very strong preference for this affix.

As described in the introduction to this chapter, Van Wijk used the Van Dale corpus for frequency data. Table 6 illustrates the phonological contexts and the affix that is dominant in this context, as observed in the Van Dale corpus. If the cell representing words with stressed final syllable ending in a sonorant holds /-en/, the Van Dale corpus tells us that this is the most frequent affix in this context.

	Stressed final syllable	Unstressed final syllable
Diphthong/glide	/-en/	/-en/
Obstruent	/-en/	/-en/
Sonorant	/-en/	/-s/
Front vowel	/-s/	/-s/
Back vowel	/-s/	/-s/

Table 6: distribution of the plural affixes in various contexts as observed in the Van Dale corpus *(from Van Wijk, 2007, p. 51)*

The relevant literature on (interaction of) stress and sonority in Dutch, most notably De Haas & Trommelen (1993), suggests that the sonority factor is stronger than the rhythmic factor. As Van Wijk explains, diphthong-final nouns take /–en/ regardless of their stress pattern. (*'kei-en', 'gewei-en', 'jaargetij(d)-en')* The same holds for obstruent final nouns. Even though the sonority and stress factors are in conflict when the final syllable is unstressed, obstruent-final nouns with an unstressed final syllable almost always take /–en/, avoiding bad sonority clusters, even though this conflicts with the factor stress. Examples are '*bierdop'* and *'scheepsvracht*', both words with an unstressed final syllable ending in an obstruent, taking '*bierdop(p)-en*' and '*scheepsvracht-en*' rather than '*bierdop-s*' and '*scheepsvracht-s*'.

For sonorant-final nouns, the sonority factor does not make a clear prediction regarding the choice between /-en/ and /–s/, because the choice between, say, '*duim-s*' and '*duim-en*' cannot be made using sonority considerations: In both configurations sonority decreases inside out per individual syllable. As table 6 shows, for words with final stress in this context /-en/ is taken, creating a trochaic stress pattern. For words without final stress in this context /-s/ is taken.

For nouns ending in a back vowel, /-s/ is the strongly preferred suffix ('*auto-s', 'opoe-s').* For nouns ending in a front vowel, Van Wijk points out that two views are found in the literature. The one that seems to hold best (De Haas & Trommelen, 1993) is the one where monosyllabic nouns ending in a front vowel(*'ree-en'*) take /–en/, and polysyllabic nouns ending in a front vowel (*'ara-s*') take /–s/ as the plural suffix.

5.3 The Constraints

As explained in the introduction of chapter 5, the purpose of Van Wijk's dissertation is to confront Pinker's dual route model with material from Dutch, including new acquisition data. However, towards the end of the thesis she also makes an attempt to model her findings within the framework of Optimality Theory. In this section, I briefly review Van Wijk's OT analysis, especially focusing on her proposed constraints. These, of course, specifically incorporate versions of Sonority and the Rhythmic factor. First, the following three basic faithfulness constraints are postulated:

1. PLURAL-REALIZE: plural must be realized.
2. MAX: every element in the input has a correspondent in the output.
3. DEP: every element in the output has a correspondent in the input.

As faithfulness constraints go, these three constraints taken together demand that what is there (covertly) should be pronounced (overtly); that everything that is present in the input should also be realized in the output and, in reverse; that everything that is realized in the output should also be present in the input.

The constraints covering the rhythmic factor are the following:

4. FTBIN: Feet are binary under syllabic analysis
5. ALIGN-WD-RIGHT: every prosodic word ends in a foot
6. RH-TYPE=T: Feet have initial prominence (are trochaic)

Together these constraints demand that words end in a binary trochaic foot: 4 demands binary feet, 5 demands a word to end in a foot and 6 demands for this foot to be trochaic.

The constraint Van Wijk proposes to model the sonority factor is the following:

7. SONSEQ: complex onsets rise in sonority and complex codas fall in sonority

For this constraint, only the part about codas is relevant. The only affix potentially creating a new onset is /–en/, which starts with a vowel and thus never creates violations of the sonority hierarchy principle. It demands that if there is any affix inserted (or, more generally anything inserted or even present at all) its onset and coda should follow the sonority principle in the specified way. To keep things constant, for all the candidate pairs it is assumed that they incur no violation of this constraint in the stem form. It would be unfair competition to assume a violation for the one stem form, but not for the other. Particularly when the focus is on the choice for plural suffix. For example, if the stem form (and the relevant plural form) of the word '*stok*' (of which the onset according to a more fine grained version of the sonority principle has a decreasing sonority (with the fricative being more sonorant than the plosive)) was to be compared to another word '*plak*', which does not have an onset decreasing in sonority, this would introduce a difference in the analysis of the two words that has nothing to do with the choice for plural affix. Therefore, I will adhere to the *ceteris paribus* assumption. The point of interest is the plural affix.

The third markedness constraint is:

8. ONSET: Syllables must have an onset

This constraint concerns words ending in a vowel[9]. At first glance, it forces words ending in a vowel to take /-s/, because /-en/ will add a new syllable without an onset. What actually happens in Dutch, is that most vowels do take /-en/, but insert a vowel in between: '*allergie-(j)-en', 'desiree-(j)-en'*. This way, this constraint remains satisfied. Obviously, at the cost of violating another; DEP, which immediately tells something about the relative ranking of these two constraints. There is, however, one vowel that 'does not do' insertion of the glide. This is the /a/. Predictably, words ending in an /a/ show a strong preference towards /-s/.

Finally Van Wijk proposes two morphological constraints, expressing the relevant suffixations:

9. PLURAL = S; plural is realized by affix /-s/
10. PLURAL = EN; plural is realized by affix /-en/

These are presented as two 'language specific instances of some universal scheme'. Recall that in the dual route model, every language should have a default value for the (plural) affix, but Van Wijk's contribution to the theoretical discussion of this model is to claim that Dutch has two such defaults: in Dutch the plural is realized by either /–s/ or /–en/. The morphological constraints make these two suffixes available at the same time, for any noun stem, and the task of the constraints, as always, is to select an optimal output candidate, which will then have one or the other suffix. See the tableaus 5 and 6 below.

input	PLURAL = S	PLURAL = EN
>> **input-s**		*
input-en	*!	

Tableau 5: Evaluation of an input given ranking PLURAL = S >> PLURAL = EN

input	PLURAL = EN	PLURAL = S
input-s	*!	
>> **input-en**		*

Tableau 6: Evaluation of an input given ranking PLURAL = EN >> PLURAL = EN

[9] For words ending on a consonant, re-syllabification can take place after adding /-en/ (which is basically an VC syllable, without an onset). '*ko-zijn-en*' will become *'ko-zij-nen'*.

In the remainder of this thesis I will use the set of constraints described in this section, to try and learn the frequency data Van Wijk constructed (by eliciting the plural form of several nonce words from adult native speakers). The learning algorithm will succeed if, based on the set of constraints and the frequency data as input, it can construct the grammar in such a way that its output will mimic the frequency distributions Van Wijk has found in the production experiment.

The frequency data I will work with, is based on Van Wijk, table 10 on page 53:

	Final	**penultimate**	**Antepenultimate**
diphthong	*68,2% /-en/* *31,8% /-s/*	*40,9% /-en/* *45,5% /-s/*	*54,5% /-en/* *31,8% /-s/*
obstruent	*93,9% /-en/* *0,0% /-s/*	*57,6% /-en/* *12,2% /-s/*	*65,9% /-en/* *13,6% /-s/*
sonorant	*87,0% /-en/* *11,7% /-s/*	*22,7% /-en/* *59,1% /-s/*	*40,9% /-en/* *31,8% /-s/*
front vowel	*51,5% /-en/* *30,3% /-s/*	*18,2% /-en/* *51,5% /-s/*	*39,4% /-en/* *48,5% /-s/*
back vowel	*36,4% /-en/* *61,4% /-s/*	*13,6% /-en/* *81,8% /-s/*	*27,3% /-en/* *63,6% /-s/*

Table 7: Frequency distributions of the suffixes in the various contexts, taken from the adult production experiments (control group)

As noted in chapter 4, for Standard OT notions such as free variation and optionality are problematic. When a grammar is defined as a certain constraint hierarchy, this grammar will produce the same output - given the same input - every time an input is under evaluation. Variation, however, implies being able to produce a different output for the same input. This would require a different constraint hierarchy, i.e., a different grammar. If we are sure that variation truly exists within a particular grammar - which is the assumption, recall chapter 4 - then this is impossible to model and learn in standard OT (as originally proposed by Tesar & Smolensky, 1993).

What is needed, then, is a single grammar/constraint hierarchy that can produce more than one output. The circumstances under which this can occur will obviously have to be very strictly specified, given empirical data such as frequency patterns found in actual usage. Several proposals have been made in the framework of OT to model such cases. I will briefly describe two alternative approaches before moving on to the introduction of the Gradual Learning Algorithm, which I will use.

6. Antilla

In chapter 4, I discussed the case of variation found in the Finnish genitive plural, as discussed in Antilla (1997). I promised to return to his actual modeling of this case, and that promise will be followed up on here. Not unlike the case of Dutch pluralisation, Antilla proposes that three factors are involved: stress, weight and sonority. The areas where these factors conflict, are the areas where variation arises.

The way Antilla derives the correct outputs, including varying ones, is through various strata. Constraints can be in a certain stratum of the hierarchy. Within this stratum, they are freely ranked with regard to each other. However, there is a strict ranking between two strata. If constraint A and B are in stratum 1, and constraints C and D are in stratum 2, then A and B are freely ranked with regard to each other, the same holds for C and D. But there is a strict hierarchy between A and B on the one hand, and C and D on the other.
This is where the variation comes from in Antilla's design. When (stratally identical) constraint A is ranked higher than B, candidate *x* will be optimal. Sometimes B is ranked higher than A, resulting in candidate *y* to be optimal. The dotted line in the tableaus 7 and 8 below indicates that B and C are in the same stratum and not ranked relative to each other. Under the first evaluation, *cand2* will be optimal, under the second *cand1* will be optimal. However, because A is in a different stratum than B and C, it will never be the case that a candidate violating A but satisfying B will be optimal at the expense of a candidate satisfying A but violating B.

	A	B	C
1 *cand1*		*	
2 >> ***cand2***			*

Tableau 7: evaluation 1 with Antilla's strata

	A	B	C
1 >> ***cand1***			*
2 *cand2*		*	

Tableau 8: evaluation 2 with Antilla's strata

As Boersma & Hayes (2001) point out, a drawback of this approach is that strata need to be quite large (containing a large number of constraints) when the distribution comes close to 99 to 1 for two candidates (as it turns out, they give the data on the dark vs. light /l/ as an example of such cases (see chapter 4, in particular the discussed cases of *'mail it'* and '*Louanne*'). In these cases, 99 possible rankings (within one stratum) need to favor the one candidate, and 1 possible ranking needs to favor the other candidate. For these cases, the number of constraints becomes quite large, and they would have to interact in quite elaborate ways.

7. Hayes

A second proposal, by Hayes (2001) serves to capture the variation found in the distribution for dark vs. light /l/ in English, as discussed in chapter 4. His proposal includes strictness bands and fringes. Within each strictness band, there is a selection point. This selection point 'defines' the 'current' value of the constraint. When the strictness bands of constraint A and B overlap, it depends on the selection point within the band for both constraints which one actually comes out on top. The selection points can vary, giving rise to variation. This can be visualized as follows (figure 1):

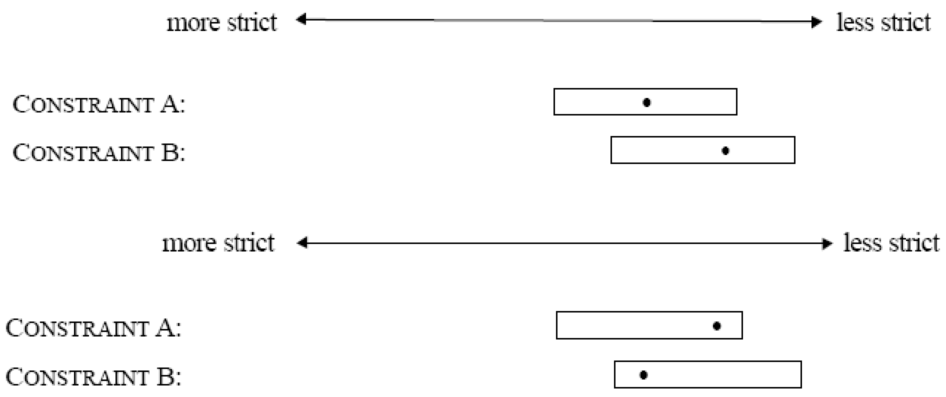

Figure 1 : Illustrations of Hayes' strictness bands and selection points (*from Hayes, 2001; p.4*)

In the upper graph, constraint A will be ranked higher than B, whereas in the lower graph, due to alternative selection points, constraint B will be ranked higher than A.

Recall that Hayes's aim with this proposal was to handle gradient well-formedness: this is where the fringes come from. The fringes are at the edges of the strictness bands. If the selection point for a constraint is in this area, the output that results will be considered odd (because it is only highly infrequently selected) but not completely out. Visualized by figure 2 below:

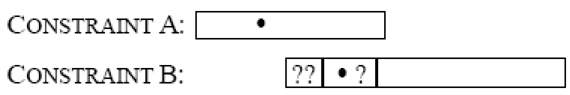

Figure 2 : illustrations of Hayes' fringes (*from Hayes, 2001; p.6*)

This is how both variation and gradient well-formedness is accounted for in this proposal. (For a full discussion, see Hayes, 2001).

The drawback of this approach, is that the arrangement of the fringes and strictness bands needs to be defined with great care. And the arrangement is in fact language specific, implying it would need a considerable amount of time for a learner to define a grammar for yet another language.

8. Introduction to the Gradual Learning Algorithm

As explained in my Introduction, the purpose of this thesis is to re-address Dutch pluralisation, and especially the variation found in the distribution of the two suffixes /–en/ and /–s/. The framework I will use to try and model this is the Gradual Learning Algorithm, proposed by Boersma & Hayes, 2001. Instead of ranking constraints using a strict hierarchy, in this approach constraints are placed on a continuous scale and can overlap with other constraints. Although this was also the case for the approach taken by Hayes (see chapter 7), the Gradual Learning Algorithm (GLA) can do without the fringes 'at the edges of the constraints' to model gradual well-formedness.

With the continuous ranking, under certain circumstances, a candidate satisfying *C3* and violating *C2* can win from a candidate violating *C3* and satisfying *C2*. So what has to be learned is the ranking of the constraints and how much overlap there should be. Although this of course follows from their ranking automatically.

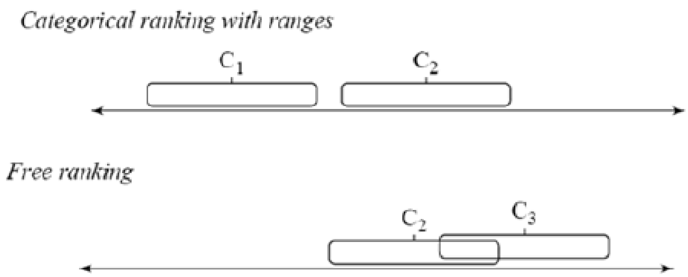

Figure 3: Strict ranking in the upper graph, free ranking in the lower graph *(from Boersma & Hayes, 2001, p. 3)*

When evaluating the possible candidates for a given input, in a continuous system apart from the ranking itself, what also has to be taken into account is the so called evaluation noise. When constraints overlap, their ranking can be reversed. In figure 1 the lower picture (*C2* and *C3*), it is usually the case that *C2* >> *C3*, but in a number of cases *C3* >> *C2*. The probability of *C3* >> *C2* happening depends on the ranking of the candidates and the evaluation noise.

The authors actually propose a normal distribution for constraints over the ranking scale, as illustrated in figure 4.

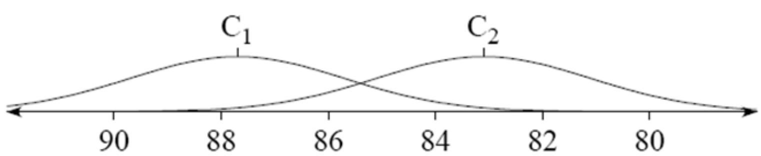

Figure 4: Value of C1 and C2. In the overlapping area, a reversed ranking C2 >> C1 is possible *(from Boersma & Hayes, 2001, p.5).*

The numbers in this figure refer to the ranking value, and are irrelevant for the example at hand. What is relevant, is that a well-known property of Gaussian distributions is that their value never really reaches zero; they have no boundaries. An important consequence is that in fact all constraints overlap. No matter how many standard deviations *C1* and *C6* (for instance) are apart, given that *C1* is ranked higher than *C6*, there is always some probability that C6 >> C1. The exact value of this probability depends on their distance to each other and the evaluation noise. Note that an important theoretical consequence of this is that the notion of grammaticality in fact disappears[10]. Because some candidate which is usually taken to be incorrect - non-optimal - always has a chance of coming out on top, one cannot say that this candidate is truly ungrammatical.

This has serious theoretical consequences[11], but, as Boersma & Hayes point out, does not really have any practical consequences. For instance, with *C1* and *C2* nine standard deviations apart, chances of a reversed ranking coming out (favouring this in most cases non-optimal candidate over the usual optimal one) are 1 in 10 billion. This practically means that one would not expect to observe this outcome even when observing a speaker for an entire lifetime. This is practically equal to

assuming the candidate that would come out in the reversed ranking is ungrammatical. Boersma & Hayes report various cases of constraints being tens or even hundreds of standard deviations apart.

In Boersma and Hayes's paper, in order to show how the approach works in concrete empirical cases, the hierarchy of a continuous ranking scale is derived from training data, trying to mimic a 'learner'. The set of constraints is taken to be known, and the data is what a learner simply runs into; the language as spoken by people around him. Apart from these two factors, there are some other parameters to be set to pursue the learning of the correct ranking. There are the following.

[10] Apart from so called harmonic bounding cases. Due to harmonic bounding, candidate B can never surface, when the violations incurred by candidate B are a subset of the violations incurred by candidate A. So it is actually still possible to absolutely exclude a candidate, thus rendering it ungrammatical.

[11] For a full discussion on this, see *Grammar without grammaticality*, Sampson, 2007.

8.1 Initial state

As a starting point, the constraints need to have some ranking. If the system will turn out to learn the data (and constraints) in a successful way, this initial ranking will not be relevant to the final ranking and outcome. There is however an important prerequisite for the system to learn the data correctly. This prerequisite is that the grammar – or ranking – that is learned converges. A grammar converges if, after some time, when fed with the relevant amount of data, that grammar does no longer change (significantly). It has arrived at the final state.

Given that the system converges, the only thing the initial ranking can do is influence the amount of data needed to arrive at the final state. But no matter what the initial state is, the final state is always the same in a converging system. For Dutch pluralisation, I will set all constraints at an initial equal and arbitrary ranking of 100.

8.2 Plasticity

In Boersma & Hayes' system, the plasticity is the size of the adjustment that is made to a constraint's ranking when encountering an error. An error here means that some constraint ranking resulted in some optimal candidate, but this candidate is not optimal in the training set. The constraints thus need to be re-ranked in order to let the correct candidate come out, i.e. the one that is also in the training set. If for example (recall introductory chapter 3) a final devoicing constraint (*Fd*) outranks a faithfulness constraint (*F*), *Fd* >> *F*, when encountering the input form */bed/*, the optimal output candidate is */bet/*, which is incorrect for English. Consequently, *Fd* has to be ranked lower or *F* has to be ranked higher. If the plasticity is 3, this means that when F was 104, it will now be 107, or when *Fd* was 105, it will now be 102.

In general, using a larger plasticity means that the system will head toward its final state relatively fast in the early state of the learning, but since it lacks in subtlety, needs a considerable amount of time to do the finetuning in the final stages. That is, if two constraints need to be a measure of 2 apart from each other (C1=89 and C2=91), this can take a lot of time if the plasticity is 10 and ranking starts at all constraints valued at 100.

To achieve the same relation, C1 now needs to be 890 and C2 910, for which ((890-89)/10=)80 steps are needed, since a constraint value can only be changed with a measure of 10 at a time.

On the other hand, using a small plasticity means that the system needs a very large amount of time in the early stages of learning, but since it can make subtle changes, has the advantage at the final stage of learning. If C1 needs to be 75, with a plasticity of 0.5 and starting at 100 for all constraints, the system will need 50 steps to achieve this state.

To combine both virtues, Boersma & Hayes have the plasticity decreasing when learning progresses, which means that they start off with a large plasticity to approach the final state relatively fast and in the end are able to make subtle changes because the plasticity is then smaller.

8.3 Variation

The main purpose of this stochastic variant of OT is that it should be capable of handling variation. This is simply simulated by making some reversed ranking possible and likely to a certain degree. If, for example, in the grammar of Dutch with optional post-schwa n-deletion *C1* >> *C2* results in the output /nEime:x@/ and *C2* >> *C1* results /nEime:x@n/, *C1* and *C2* are probably going to be ranked relatively close together. In combination with the evaluation noise, they will mimic the frequency patterns of the two possible realizations. If the variant with the final sonorant occurs in 70% of the cases, *C2* will come to outrank *C1* in 70% of the cases.

Note that in this approach, optionality, gradient well-formedness and variation are all handled in the same manner: to the system, they are equal. All are cases of one input form having more than one possible surface form.

8.4 Robustness

As a result of its design, stochastic OT is more robust than standard OT in its learning process. Boersma & Hayes discuss the case of an error in the training set: when standard OT comes across an error, it will, in order to let the correct surface form come out on top, need to reverse the ranking of the responsible constraints. This reaction is of course not correct, since we are dealing with an error.

Consequently, when the standard OT system after some time runs into the correct surface form of this given input form again, it will flip its ranking again. Stochastic OT on the other hand will only slightly adjust its ranking when encountering a speech error. Many speech errors are therefore needed to really have some effect on the ranking. This means that the stochastic variant is more robust. In fact, speech errors are simply considered as variants of some input form, which is basically the same again as free variation and optionality, the only difference being that errors are much less frequent.

Eventually, all of the advantages of Boersma & Hayes's stochastic OT variant are the result of its ability to handle variable outputs, e.g. not always the same surface form for a given input. In fact it just captures the distribution of the data in frequency patterns and this is exactly what is learned.

9. Learning the data

We have arrived at the point in this thesis where I can now readdress the case of Dutch pluralisation and its variation, as discussed by Van Wijk (2007). I will use the data taken from Van Wijk as already given in chapter 5, and try to show how the Boersma & Hayes' approach can be applied to it. I will use Van Wijk's constraints and try to find out whether GLA is able to reproduce the training data.

9.1 Defining and learning the constraints

One of Boersma's major practical contributions to recent linguistics is the Praat programme (freely available from: http://www.fon.hum.uva.nl/praat/). This programme is mainly used for purposes of phonetic analysis, but also has an easy to use built-in version of the GLA. In a simple script, the user can define the constraints in the given syntax. Then the input-output pairs and their frequency distributions have to be defined. If these two scripts are taken together, Praat with its built-in learning algorithm can 'learn' the data and comes up with a constraint ranking for the constraints defined.

The constraints in the script are solely defined by the candidates they either violate or satisfy. In Praat, there is no inherent way of defining the nature or purpose of the constraints.

The constraints (the first four) are initiated as follows:

```
"P\s{lural-Realise}"    100 100 1          ! Plural must be realised
"M\s{ax}"               100 100 1          ! Input corr. Output
"D\s{ep}"               100 100 1          ! Output corr. Input
"F\s{tBin}"             100 100 1          ! Feet are binary
```

Here, the string in quotation marks is the name of the constraint (the slashes and accolades are to define the fonts). The first number is the initial ranking (see chapter 3).
The second number is the initial disharmony. The disharmony indicates 'how sure' the system is of its actual rankings. If the ranking value and the value for disharmony are far apart, it can be concluded that the learning procedure has not resulted in a correct grammar.
The third number is the plasticity; the value by which the ranking is changed if re-ordering of the constraints should take place (see chapter 8.2). (Everything after "!" on the line are comments, not read by the system (only for the readability of the user))

Given this, the definition of the constraints depends on the input-output correspondence:

```
"DiphtPen" 2
      "DiphtPen-s" 0 0 0 0
      "DiphtPen-en" 0 0 1 1
```

Here, the string between quotation marks is the input candidate and refers to the pair distributions script. The string between quotation marks on the second line is the first candidate. The numbers

following refer to the constraints in chronological order. The first “0” means no violations for the constraint PLURAL-REALIZE, the third “1” means one violation for the constraint DEP.
The third line holds the second candidate.
Since only the two plural suffixes of Dutch are relevant here, no more candidates are present, but the list can be as long as needed.

The pair distribution syntax is as follows:

```
"DiphtPen" "DiphtPen-s" 52.7
"DiphtPen" "DiphtPen-en" 47.3
```

The first string between quotation marks holds the input candidate which is referred to from the constraints script. The second string between quotation marks holds the output - which is also present in the constraints script - corresponding to this input. The number that follows is the frequency. (See the appendix for the complete scripts used.)

9.2 Results

In this section I will present the results of my attempt to learn the frequency data from Van Wijk regarding Dutch pluralisation with the GLA.
But first consider a number of prerequisites. I assume that with respect to applying stress patterns, plural nouns cannot be treated individually. They inherit the stress pattern of the corresponding singular form. When possible, the /–en/ suffix can be incorporated into an existing foot already, to form a binary and trochaic foot. But adding the plural suffix does not have any effect on the foot structure and thus on the stress pattern of the rest of the word. For example, in the case of *‘kapitein’* taking
/-en/, the affix will be incorporated with the second foot -which is in the singular form a monosyllabic foot - and which will become the foot ‘*teinen*’. But in the case of ‘*baken’* taking /-en/, the stress pattern of the singular form will prevail. If ‘*bakenen’* as an individual word is to be ‘footed’, the more logic division would be ‘*kenen’* (the second and third syllable) as a foot. However, because the singular form has *‘baken’* (the first and second syllable) as a foot, this is what the plural form will inherit.

In addition, it can be confusing to look at the stem forms as such. OT only scores the output candidates relative to each other, and I am only interested in the choice of plural suffix for a given noun stem. For example, if there is some violation of the sonority sequence constraint in the onset of the stem, it is there in both the /–en/ and /–s/ candidate. They would still score equal relative to each other on this point.

For nouns with final stress, when the /–en/ plural suffix is added, a binary and trochaic foot is formed. Obviously, in case of the /–s/ plural suffix, nothing happens with regard to foot structure.
For nouns with penultimate stress, I assume that the singular form of the noun ends with a trochaic foot. In case of the /–s/ plural suffix, nothing happens. When the /–en/ plural suffix is taken, the word does not end in a foot anymore. No restructuring of the feet in the word takes place; the /–en/

suffix remains non-footed (as explained above).
For nouns with antepenultimate stress, I assume that the singular form of the noun ends with a non-footed syllable. In case of the /-s/ suffix, nothing happens. When the /–en/ suffix is taken, the final non-footed syllable and the suffix /–en/ form a new trochaic foot together. No restructuring takes place, only the addition of a new foot, which does not bare stress.

Unlike real subjects, Praat does not care about the actual form the test cases take – this as long as the constraints are defined correctly, also in combination with the frequency distribution data.

I have taken the above rules and defined the constraints in the following way:

```
"ObsFin" 2
      "ObsFin-s" 0 0 0 1 0 1 1 0 0 1
      "ObsFin-en" 0 0 0 0 0 0 0 0 1 0
```

The first candidate is 'some noun with final stress and ending on an obstruent' which takes the /–s/ plural suffix, followed by the violations of the relevant constraints.
The second candidate is 'some noun with final stress and ending on an obstruent' which takes the /–en/ plural affix, followed by the violations of the relevant constraints.

The clear advantage of this approach in comparison to picking particular example nonce words is that there is no way in which the example itself can have any meaning and influence the outcome.
This would for example be the case if the 'Sonorant Penultimate' candidate was /dOktOr/ or /mOtOr/. These are exceptions in which the stress shifts from the first syllable (/dOk/) in the singular form, to the second syllable (/tOr/) in the plural form. This stress shift is a known exception for nouns ending on /-Or/.
Again, to avoid undesirable effects, I have assumed some hypothetical noun with the relevant properties (for stress and final phoneme).

Before moving on to the results, it is important to note some other assumptions I have made.

For the nouns ending in a vowel - either front, back or diphthong - I have assumed the insertion of a glide between the stem form and the suffix. This is what speakers of Dutch naturally do[12], except for when this vowel is an /a/. See chapter 5.3 for an explanation

The constraint DEP works on phonemes (only). If some syllable is footed in the output, and not footed in the input, this does not incur a violation of the constraint output-input correspondence.

[12] Which would seem to imply that ONSET is ranked higher than DEP in Dutch.

Given the following constraints:

1. PLURAL-REALIZE – states that the plural must be realized.
2. MAX – states that every element in the input should be present in the output.
3. DEP – states that every element in the output should be present in the input.
4. FTBIN – states that if there is a foot in the output, it should be binary.
5. ALIGN-WD-RIGHT – states that the final syllable of a word should be footed.
6. RH-TYPE=T – states that if there is a foot in the output, it should be trochaic.
7. SON-SEQ – states that every onset in the output should rise in sonority and every coda in the output should fall in sonority.
8. ONSET – states that every syllable in the output should have an onset.
9. PLURAL-S – states that the plural should be realized by the plural suffix /–s/.
10. PLURAL-EN – states that the plural should be realized by the plural suffix /–en/.

In Praat, I have used the following settings when learning the constraints given the frequency distributions.

Evaluation noise: 2.0
Reranking strategy: Symmetric all
Initial plasticity: 1.0
Replications per plasticity: 1.000.000
Plasticity decrement: 0.1
Number of plasticities: 4
Relative plasticity spreading: 0.1

I fed the learning algorithm with a total of 4.000.000 (four million) pairs of learning data.

For the grammar to be learned, I then generated output distributions, using 10.000 trials per input, with again an evaluation noise of 2.0.

This resulted in the following ranking and the following output distributions.

	Ranking value	Disharmony	Plasticity
PLURAL-S	100.131	101.913	1.000
ONSET	100.000	100.545	1.000
MAX	100.000	98.919	1.000
FTBIN	100.195	98.900	1.000
SON-SEQ	103.221	98.717	1.000
RH-TYPE=T	100.195	98.311	1.000
ALIGN-WD-RIGHT	99.496	97.488	1.000
DEP	99.802	97.464	1.000
PLURAL-REALIZE	100.000	96.995	1.000
PLURAL-EN	99.869	96.774	1.000

Table 8: Constraint hierarchy as learned by the GLA

Table 8 shows that the ranking values are all very close together. Some constraints even have the same ranking and/or still their initial ranking. Also, some constraints have a large disharmony.

	Expected Frequency	Normalized Expected Frequency (NEF)	Results 4m	Results 400k	NEF – Results 4m	NEF – Results 400k	Avg. error 4m	Avg. error 400k	% lost
DiphtFin-s	31,8	31,8	38,0	37,8	6,2	6,0	10,38	10,46	0,0
DiphtFin-en	68,2	68,2	62,0	62,2	6,2	6,0			0,0
DiphtPen-s	45,5	52,7	74,4	73,7	21,7	21,0			13,6
DiphtPen-en	40,9	47,3	25,6	26,3	21,7	21,0			13,6
DiphtAntepen-s	31,8	36,8	55,1	53,8	18,3	17,0			13,7
DiphtAntepen-en	54,5	63,2	44,9	46,2	18,3	17,0			13,7
ObsFin-s	0	0	8,4	8,4	8,4	8,4			6,1
ObsFin-en	93,9	100	91,6	91,6	8,4	8,4			6,1
ObsPen-s	12,1	17,4	16,5	16,9	0,9	0,5			30,3
ObsPen-en	57,6	82,6	83,5	83,1	0,9	0,5			30,3
ObsAntepen-s	13,6	17,1	10,4	10,4	6,7	6,7			20,5
ObsAntepen-en	65,9	82,9	89,6	89,6	6,7	6,7			20,5
SonFin-s	11,7	11,9	25,3	25,6	13,4	13,7			1,3
SonFin-en	87,0	88,1	74,7	74,4	13,4	13,7			1,3
SonPen-s	59,1	82,2	66,0	65,7	6,2	6,5			18,2
SonPen-en	22,7	27,8	34,0	34,3	6,2	6,5			18,2
SonAntepen-s	31,8	43,7	39,6	38,8	4,1	4,9			27,3
SonAntepen-en	40,9	56,3	60,4	61,2	4,1	4,9			27,3
FrontFin-s	30,3	37,0	54,6	53,7	17,6	16,7			18,2
FrontFin-en	51,5	63,0	45,4	46,3	17,6	16,7			18,2
FrontPen-s	51,5	73,9	74,4	73,7	0,5	0,2			30,3
frontPen-en	18,2	26,1	25,6	26,3	0,5	0,2			30,3
FrontAntepen-s	48,5	55,2	54,8	53,3	0,4	1,9			12,1
FrontAntepen-en	39,4	44,8	45,2	46,7	0,4	1,9			12,1
BackFin-s	61,4	62,8	37,9	37,7	24,9	25,1			2,2
BackFin-en	36,4	37,2	62,1	62,3	24,9	25,1			2,2
BackPen-s	81,8	85,6	74,4	73,8	11,2	11,8			4,6
BackPen-en	13,6	14,4	25,6	26,2	11,2	11,8			4,6
BackAntepen-s	63,6	70,0	54,8	53,5	15,2	16,5			9,1
BackAntepen-en	27,3	30,0	45,2	46,5	15,2	16,5			9,1

Table 9: Results

In table 9 - the results - the first column contains the candidates and the column ‘expected’ contains the precise frequency data collected by Van Wijk, given earlier in chapter 5.

The next column holds the Normalized Expected Frequency. In Van Wijk’s survey, some of her subjects output plural suffixes different from /–en/ or /–s/. The sum of frequencies for both /–s/ and /–en/ does therefore not always equal 100. This is problematic for the comparison with the output of the GLA. Since the GLA will only have two candidates - /-s/ and /–en/ - to choose from, it will distribute the output in such a way that both will equal 100. This would introduce an a priori difference between the GLA output and Judith’s observed frequency data. If I am to score the GLA on ‘the Dutch plural’, this is not a fair comparison. This is why I normalized Van Wijk’s frequencies, so that the two options do sum up to 100. This is done in the following way; for ObsPenultimate for example, /-s/ has a 12.1 score and /–en/ has a 57.6 score. This adds up to 69,7. To normalize the score for /–s/: 12,1 / (12,1+57,6) * 100 = 17,36. The score for /–en/: 57,6 / (12,1+57,6) * 100 = 82,64. And 17,36 + 82,64 = 100.

The fifth column contains the results for the run with 4.000.000 trials.
The sixth column contains the results for the run with 400.000 trials (discussed later in this chapter).
The seventh column contains the average over all errors for the run with 4.000.000 trials.
The eighth column contains the average over all errors for the run with 400.000 trials.
The ninth and final column contains the percentage that is ‘lost’, as explained above (subjects selecting suffixes other than –en or –s).

As can be seen in table 9, for some instances the GLA output closely resembles the normalized expected frequency. For the Front vowel-Antepenultimate, there is a difference of 0.4. For the Front vowel-Penultimate there is a difference of 0.5 and for Obstruent-Penultimate a difference of 0.9.

However, for some cases the difference between normalized expected frequency and GLA output is considerable. For Back vowel-Final it is 24,9, for Diphthong-Penultimate it is 21.7 and for Diphthong-Antepenultimate it is even 18,3.

One of the supposed advantages of the GLA, is that it should be able to model a large variety of frequency distributions. Whereas Antilla’s stratal grammars have difficulty (or need a large number of constraints in one stratum) producing a 90/10 distribution, this is advantageously different for the GLA.

When taking a closer look at the results, it appears that it indeed cannot be the frequency distributions that account for the large differences between normalized expected frequency and GLA output.

For Diphthong-final, the difference between normalized expected frequency and GLA output is 6, with /–s/ and /–en/ having a distribution of, respectively, 31.8 and 68.2.

For Diphthong-antepenultimate, the difference between normalized expected frequency and GLA output is 18.3. For /–s/ and /–en/ this form has a distribution of 36.8 and 63.2. This closely resembles the distribution for Diphthong-final, but with a significant difference in the error.

The same holds for Back vowel-final compared to Diphthong-antepenultimate. Back vowel-final has an error of 24.9 and for /–s/ and /–en/ a distribution of, respectively, 62.8 and 37.2, and Diphthong-antepenultimate has an error of 18.3 and for /–s/ and /–en/ a distribution of, respectively, 36.8 and 63.2. Note that the fact that the distributions are flipped (for /–s/ and /–en/) is *in this respect* of no importance to the GLA; it 'just' learns frequency patterns.

In conclusion, it is not the frequency patterns themselves that are hard to learn for the GLA. Moreover, almost every possible output, in general, is produced. A near-50/50 distribution is produced for Front vowel-antepenultimate (45.2/54.8). At the other end, closest to 0/100 is Obstruent-final, with a distribution of 8.4/91.6. In-between distributions are also present.

Although the GLA is obviously able to produce almost anything in between a 50/50 and 0/100 distribution, there are some interesting things going on regarding the frequency patterns in the output. It looks like there is a tendency to, for a given distribution, systematically output (another) distribution.

In those cases in which there is a normalized expected frequency distribution of 30/70 (or around), the GLA seems to output more of a 50/50 distribution. This is the case for Back-antepenultimate (a 70/30 expected becomes 45.2/54.8) and for Diphthong-antepenultimate (a 36.8/63.2 expected becomes 44.9/55.1).

In the case of Diphthong-penultimate, the opposite seems to happen. Diphthong-penultimate has a normalized expected frequency distribution of 52.7/47.3, the GLA outputs 74.4/25.6.

In the case of Back vowel-final, the normalized expected frequency distribution of 62.8/37.2 is flipped and becomes 37.9/62.1.

In the case of Sonorant-final, the expected 11.9/88.1 becomes 25,3/74,7. It cannot be concluded that the GLA therefore has a preference towards a 50/50 distribution, because the Obstruent-antepenultimate case has an expected 17.1/82.9 distribution, which becomes 10.4/89.6, sliding the other way.

The significant errors made by the GLA are not the result of certain frequency distributions that might be hard to learn, since it does output almost everything in between 50/50 and 100/0.

Note that the last results column of table 9, i.e. the one giving the lost percentages (the number of times subjects chose for another suffix than either /–s/ or /–en/) is only indicative for the subjects themselves. It shows how often they selected either of the default suffixes. The GLA, while learning, only uses the normalized frequency data and does not know the original percentages. It does not know how often a suffix other than one of the defaults is chosen. For the Sonorant-Final context, in only 1.3% of the cases, subjects selected a suffix different from the defaults - whereas the GLA was

13,4 % off in this context. The other way around, for the Obstruent-Penultimate case, in as many as 30,3% of all cases subjects selected a different suffix than one of the defaults. Here, the GLA was only 0,9% off in its prediction.

Perhaps the significant errors that the GLA makes have their source in the constraint violation patterns. The Diphthong-final candidate, with an error of 6, shows the following pattern:

```
"DiphtFin" 2
      "DiphtFin-s" 0 0 0 1 0 1 0 0 0 1
      "DiphtFin-jen" 0 0 1 0 0 0 0 0 1 0
```

Giving DiphtFin–s one violation for the Binary Feet constraint because this form has final stress - I have assumed a super-heavy final syllable forming a foot on its own – and also one violation for the 'Trochaic Feet' constraint (for the same reason) and the 'Plural realized by /–en/' constraint (obviously because here the plural is not realized by /–en/).
For DiphtFin-jen, the candidate has one violation for the output-input correspondence constraint because of the added glide, and one violation for the Plural realized by /–s/ constraint, obviously because here the plural is not realized by /–s/.

Note that Back-fin has exactly the same violation pattern for both candidates, and produces almost the same output (371.9 for /–s/ and 62.1 for /–en/). Here however, the frequencies are flipped (the actual distribution should be 62.8 for /–s/ and 37.2 for /–en/, as found by Van Wijk), resulting in the largest error found; 24.9.

The underlying problem that the examples Dipht-Fin and Back-fin above illustrate is very problematic for the GLA: some pairs that have the same violation patterns go together with very different frequency distributions.
The constraints are ranked on the basis of the violation patterns and the normalized frequency distributions that go with the pairs. For two different pairs, it could well be the case that the violation pattern is the same. Consider:

```
"DiphtFin" 2
      "DiphtFin-s" 0 0 0 1 0 1 0 0 0 1
      "DiphtFin-jen" 0 0 1 0 0 0 0 0 1 0
```

It would be hard to argue that the violation pattern should be any different. For the /–s/ candidate, there is one violation for FtBin, because, as noted earlier, I assumed that the final syllable - which bears main stress - has to form a foot on its own. The more complicated option would be to assume a iambic foot. That way, Rh-Type=T is still violated, but at least FtBin is satisfied. The violation for Plural-en is straightforward.
Turning to the /–en/ candidate, I assumed insertion of a glide, because this is what actually happens in Dutch. One could try to argue against these specific choices.

However, this choice for the one way or the other is irrelevant for the case at hand. Consider:

```
"BackFin" 2
      "BackFin-s" 0 0 0 1 0 1 0 0 0 1
      "BackFin-jen" 0 0 1 0 0 0 0 0 1 0
```

The case of Back-fin shows exactly the same violation pattern as Dipth-fin, independently of assumptions about the insertion of a glide or not, or either a foot formed by one super-heavy syllable or an iambic foot. The violation pattern of both pairs would still be the same.

The conclusion seems to be inescapable that this is a case of two pairs having exactly the same pattern, but a different distribution as found by Van Wijk. This is impossible to learn for the GLA, since the violation pattern of the candidates is the only clue the GLA has to make distinctions. When looking at the output, it seems as if the GLA takes one of the possible frequency distributions for a possible violation pattern, and conforms to that one. Another option would be to average all frequency distributions for a given violation pattern, and converge to that point. But this is not what happens.

Recall from chapter 8.1 the explanation of a converging system. As an additional experiment, I ran the GLA again, but now with different settings. Instead of four million input trials, I presented it with four hundred thousand input trials, to see if the results the GLA then comes up with, differs from the results of the four million trial. The results are presented in columns six and eight of table 9.

As table 9 shows, the distributions output by the GLA (and hence the average error) do not differ significantly. However, the ranking the algorithm produces, is scrambled in comparison to the ranking for the four million trial.

When the grammar is learned using 100.000 replications per plasticity and 4 plasticities, resulting in 400.000 trials, the following ranking emerges:

	Ranking value	Disharmony	Plasticity
DEP	99.739	104.388	1.000
SON-SEQ	103.182	101.984	1.000
ONSET	100.000	101.903	1.000
RH-TYPE=T	100.129	100.512	1.000
PLURAL-S	100.103	99.760	1.000
PLURAL-REALIZE	100.000	99.333	1.000
ALIGN-WD-RIGHT	99.525	98.913	1.000
MAX	100.000	98.801	1.000
PLURAL-EN	99.897	98.778	1.000
FTBIN	100.129	97.846	1.000

Table 10: Hierarchy with 400k trials

As shown earlier, the ranking with four million trials was:

	Ranking value	Disharmony	Plasticity
PLURAL-S	100.131	101.913	1.000
ONSET	100.000	100.545	1.000
MAX	100.000	98.919	1.000
FTBIN	100.195	98.900	1.000
SON-SEQ	103.221	98.717	1.000
RH-TYPE=T	100.195	98.311	1.000
ALIGN-WD-RIGHT	99.496	97.488	1.000
DEP	99.802	97.464	1.000
PLURAL-REALIZE	100.000	96.995	1.000
PLURAL-EN	99.869	96.774	1.000

Table 11 (table 8 from chapter 9.2): Hierarchy with 4m trials

The conclusion already follows, of course, from the actual ranking values and disharmony, i.e. that the GLA is not sure of its conclusions. But the fact that the difference in hierarchy between 4m and 400k trials is remarkable, and the difference between the actual results (in output distributions) between 4m and 400k trials is not significant, is even more evidence that the constraints are very close together. As a result the outcomes sometimes vary excessively widely.

10. Evaluation and Discussion

In their paper Boersma & Hayes report an average percentage of errors of 2.54% for the Finnish genitive plural case and even 0.17% for the data on dark vs. light /l/. Given the current error average of about 10% for the Dutch pluralisation case, the obvious question is: What could be the reason for the high number of errors by the GLA in its output distributions for the Dutch plural affix for nouns?

As shown in Boersma & Hayes and in my own results, it is not the case that the GLA is unable to deal with the respective frequency distributions present in the Dutch plural affix data.
The problem is that the only clue the GLA has is the violation pattern of some candidate. Given two pairs with identical violation patterns, but with different frequency distributions, it now has to make a distinction. This looks more like guessing: there just is not enough information to make a well-argued decision.

In this light, one might be tempted to conclude that the key to success lies in having a unique violation pattern. This, however, does not appear to be true. As it turns out, the only six contexts that do have a unique violation pattern are the Obstruent-Final, the Obstruent-Penultimate, the Obstruent-Antepenultimate, the Sonorant-Final, the Sonorant-Penultimate and the Sonorant-Antepenultimate. They have very diffuse error values; respectively 8,4%, 0,9%, 6,7%, 13,4%, 6,2% and 4,1%.

One possibility that comes to mind is that the constraints used in my analysis, which – the reader will recall – were simply adopted from Van Wijk - were after all inadequate for a fruitful Stochatisc O.T. analysis of the Dutch plural suffix. Note that neither Van Wijk nor I have used the method of factorial typology (see chapter 1.3) on the set of constraints. Although I am not sure whether this would make a difference (e.g. if the GLA 'cares about' the constraints being universal or not), it certainly is a suggestion for further research.

In chapter 7.3 of Van Wijk, the constraints are able to predict the areas where variation occurs. It is to be expected that, where more constraints favor one outcome over the other, the percentages (regarding the distribution of the two possibilities) follow this pattern. However, the true percentages come from the subjects, and not (yet) from the constraints. A reason for the difference between expected frequency based on the subjects' judgments and the GLA output could be an artifact of the data Van Wijk used when testing the subjects. If, accidentally, something was going on there that was not covered by the constraints, this could perfectly explain the errors in the GLA predictions. An example of such an artifact are Dutch nouns ending in /-or/ (discussed in chapter 9.2). These are exceptions to the 'normal' pattern. If the data the subjects had to judge contained these types of exceptions, this means that more was going on than just the stress and sonority factor covered by the constraints.

In any case, it is obvious that the ranking resulting from my analysis is 'unstable'. Given the large disharmony values, it is easy to see that the GLA is not very sure of its hierarchy. Because of the nature of the algorithm, having constraints ranked close together means having variable outputs. On the basis of these data, the GLA does not seem to be able to get its constraints 'moving away enough' from each other. The sheer amount of data, however, should not be the problem. For the dark vs. light /l/ case discussed by Boersma & Hayes, the number of constraints is 10, with only 7 input-output pairs. And for the Finnish genitive plural, they assume 10 constraints with 22 input-output pairs. These numbers are even smaller than the numbers in my experiment; 10 constraints with 30 input-output pairs.

Given all of these considerations, to try and clear the field a bit, I decided to run a test omitting the completely unviolated constraints (which are Plural-Realize, Max, Son-Seq and Onset).

As these constraints are never violated, they do not (by definition) prefer one candidate over the other. Therefore, leaving them out should have no effect on the final outcome. With these four constraints eliminated, I arrived at an error average of 11,05%. This figure differs over 0.5% from the original results. When this difference is too high to be caused by the stochastic element of the system, and therefore something meaningful is going on, I have no explanation for what this something meaningful might be.

Because the violation patterns all look terribly alike (as discussed, only six of the fifteen candidates have a unique pattern), perhaps one of the constraints, namely the Son-Seq constraint, could be restated in a more 'gradual' way. Currently it provides a candidate with a mark when its consonantal cluster does not obey the sonority hierarchy. It could be redefined in such a way that it gives one mark for every 'stratum' the candidate is off - the strata, for example, being plosives, fricatives, glides. That way, two different contexts currently having an equal violation pattern, can be kept separate, resulting in more diversity and perhaps a better ranking.

In addition to the results and alternative procedures sketched so far, I have experimented to some extent with either the ranking or the output candidates.

Under larger plasticity the constraints will be ranked further apart. Therefore, I ran the system with all things equal except the plasticity: 10 instead of 1. The resulting ranking and output distribution did not show any tendency - let alone the correct one - at all. Using a smaller plasticity of 0.1 instead of 1, the results were more or less the same; an average error of 10,43 instead of 10,46, which is probably due to the stochastic nature of the system.

As discussed in chapter 3, OT thinking about acquisition prohibits an initial state with the faithfulness constraints outranking the markedness constraints. This can easily be simulated in Praat, because the initial ranking can be specified. When trying to have the same grammar (data) learned, but with faithfulness initially outranking markedness, the actual result was an error average of 11,05%, which is just about 0,5% higher than the original run. As will be explained by any introduction to OT, this initial configuration should be impossible to learn. This little experiment seems to suggest otherwise, at least for the current case, because the outcome is not very far off the original outcome.

However, when taking a closer look at the actual constraints involved, it becomes clear that they are never violated at all - thus showing no preference for one candidate over the other. It can therefore still be concluded that it is impossible to learn a natural language with this initial configuration, but this particular case just does not show the difference.

The other way round, with the markedness constraints outranking the faithfulness constraints in initial stage, resulted in an average error of 10,85%. This is closer, but not significantly, to the results of the original run.

11. Conclusion

The aim of this thesis was to test the GLA on the data of the Dutch pluralisation of nouns.
By providing the GLA with a set of constraints – which were taken from Van Wijk and based on the two factors stress and sonority, determining the distribution of the plural affix – and feeding it with the Dutch pluralisation data, the test case was to learn the correct grammar to produce/predict the plural of Dutch nouns correctly.

The results – of the main- and several alternative experiments - showed a much larger average error than the errors in the tests Boersma & Hayes ran themselves, suggesting that the GLA is not able to learn this variation in Dutch pluralisation.
However, as discussed in chapter 10, I think the reason for the failure in learning the Dutch data is not the design of the GLA itself, but the constraints I used when trying to learn the data.
Therefore, the most valuable suggestion for further research is to redefine the constraints – according to the suggestions done in chapter 10 - and repeat the experiment.

12. References

Anttila, A. 1997. *Deriving variation from grammar: A study of Finnish genitives*. In *Variation, change and phonological theory*, ed. Frans Hinskens, Roeland van Hout, and Leo Wetzels. Amsterdam: John Benjamins.
Baker, C.l. 1979, Syntactic theory and the projection problem, Linguistic Inquiry 10, 233-280

Boersma, P. 1997. *How we learn variation, optionality, and probability*. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* 21:43–58

Boersma, P. 2004. *A Stochastic OT account of paralinguistic tasks such as grammaticality and prototypicality judgments*. Ms University of Amsterdam

Chomsky, N. *New horizons in the study of language and mind.* 2000. Massachusetts Institute of Technology

De Haas, W & Trommelen, M. 1993. *Morfologisch handboek van het Nederlands; een overzicht van de woordvorming*. 's-Gravenhage: SDU

Goldwater, S. and Johnson, M. 2003. *Learning OT constraint rankings using a maximum entropy model.* In J. Spenader, A. Eriksson, and Ö. Dahl (eds), Proceedings of the Stockholm Workshop on Variation within Optimality Theory, p. 111-120. Stockholm University, Stockholm.

Hayes, B. 2001. *Gradient well-formedness in Optimality Theory*. In *Optimality Theory: Phonology, syntax, and acquisition,* ed. Joost Dekkers, Frank van der Leeuw, and Jeroen van de Weijer. Oxford: Oxford University Press

Houet, H. *Kramers Stijlgids Nederlands.* 2002. Utrecht: Het Spectrum

Jäger, G. 2003. *Maximum Entropy Models and Stochastic Optimality Theory.* In J. Grimshaw, J. Maling, Ch. Manning, J. Simpson, and A. Zaenen (eds.), *Architectures, Rules, and Preferences: A Festschrift for Joan Bresnan,* CSLI Publications, Stanford

Keller, F. and Ash A. 2002. *Probabilistic Learning Algorithms and Optimality Theory.* Linguistic Inquiry 33 (2), p. 225-244

Levenshtein, V.I. *Binary codes capable of correcting deletions, insertions, and reversals*. Soviet Physics Doklady, 1966

Lombardi, L. *Positional faithfulness and voicing assimilation in Optimality Theory*. 1999. Natural Language and Linguistic theory 17

McCarthy, J. and Prince, A. 1994. *The emergence of the unmarked: optimality in prosodic morphology*. *Papers of the 24th Annual Meeting of the North Eastern Linguistic Society*, ed. Mercè González, 333–379. Amherst: Graduate Linguistic Student Association

Pater, J. 2008. *Gradual learning and convergence.* Linguistic Inquiry 39 (2), p. 334-345

Prince, A. and Smolensky, P. 1993. *Optimality Theory: Constraint interaction in generative grammar*. Rutgers University Center for Cognitive Science Technical Report 2

Sampson, G.R. *Grammar without grammaticality.* 2007. Special issue of Corpus Linguistics and Linguistic Theory 3 (1)

Tesar, B. and Smolensky, P. 1993. *The learnability of Optimality Theory: An algorithm and some basic complexity results*. Boulder, University of Colorado

Van Wijk, J. 2007. *The acquisition of the Dutch plural*. Utrecht, LOT

Zonneveld, W. 1996 *Optimality Theory: Roots and aims*. South African Journal of Linguistics, Supplement 31, 23-54.

13. Appendix

“constraints Plural Landscape Adults equal ranking”

```
"ooTextFile"
"OTGrammar 3"
<OptimalityTheory>
0.0                                   ! leak
10                                    ! number of constraints

"P\s{lural-Realise}"  100 100 1       ! Plural must be realised
"M\s{ax}"             100 100 1       ! Input corr. Output
"D\s{ep}"             100 100 1       ! Output corr. Input
"F\s{tBin}"           100 100 1       ! Feet are binary
"A\s{lign-Wd-Right}"  100 100 1       ! Prosodic word ends in foot
"R\s{h-Type=T}"       100 100 1       ! Feet are trochaic
"S\s{on-Seq}"         100 100 1       ! Complex onsets rise, complex codas fall in sonority
"O\s{nset}"           100 100 1       ! Syllabes must have an onset
"P\s{lural-s}"        100 100 1       ! Plural is realised by s
"P\s{lural-en}"       100 100 1       ! Plural is realised by en

0 ! number of fixed rankings
15 ! number of accepted inputs

"DiphtFin" 2 ! input form with number of output candidates; Singular noun ending on diphthong
with stress on final syllabe
        "DiphtFin-s" 0 0 0 1 0 1 0 0 0 1! first candidate with violations
        "DiphtFin-jen" 0 0 1 0 0 0 0 0 1 0! second candidate with violations
"DiphtPen" 2
        "DiphtPen-s" 0 0 0 0 0 0 0 0 0 1
        "DiphtPen-jen" 0 0 1 0 1 0 0 0 1 0
"DiphtAntepen" 2
        "DiphtAntepen-s" 0 0 0 0 1 0 0 0 0 1
        "DiphtAntepen-jen" 0 0 1 0 0 0 0 0 1 0
"ObsFin" 2
        "ObsFin-s" 0 0 0 1 0 1 1 0 0 1
        "ObsFin-en" 0 0 0 0 0 0 0 0 1 0
"ObsPen" 2
        "ObsPen-s" 0 0 0 0 0 0 1 0 0 1
        "ObsPen-en" 0 0 0 0 1 0 0 0 1 0
"ObsAntepen" 2
        "ObsAntepen-s" 0 0 0 0 1 0 1 0 0 1
        "ObsAntepen-en" 0 0 0 0 0 0 0 0 1 0
"SonFin" 2
        "SonFin-s" 0 0 0 1 0 1 0 0 0 1
        "SonFin-en" 0 0 0 0 0 0 0 0 1 0
"SonPen" 2
        "SonPen-s" 0 0 0 0 0 0 0 0 0 1
        "SonPen-en" 0 0 0 0 1 0 0 0 1 0
"SonAntepen" 2
        "SonAntepen-s" 0 0 0 0 1 0 0 0 0 1
        "SonAntepen-en" 0 0 0 0 0 0 0 0 1 0
"FrontFin" 2
        "FrontFin-s" 0 0 0 0 1 0 0 0 0 1
        "FrontFin-jen" 0 0 1 0 0 0 0 0 1 0
"FrontPen" 2
        "FrontPen-s" 0 0 0 0 0 0 0 0 0 1
        "FrontPen-jen" 0 0 1 0 1 0 0 0 1 0
"FrontAntepen" 2
        "FrontAntepen-s" 0 0 0 0 1 0 0 0 0 1
        "FrontAntepen-jen" 0 0 1 0 0 0 0 0 1 0
"BackFin" 2
        "BackFin-s" 0 0 0 1 0 1 0 0 0 1
        "BackFin-jen" 0 0 1 0 0 0 0 0 1 0
"BackPen" 2
        "BackPen-s" 0 0 0 0 0 0 0 0 0 1
        "BackPen-jen" 0 0 1 0 1 0 0 0 1 0
"BackAntepen" 2
        "BackAntepen-s" 0 0 0 0 1 0 0 0 0 1
        "BackAntepen-jen" 0 0 1 0 0 0 0 0 1 0
```

“pair distributions Plural landscape adults”

```
"ooTextFile"
"PairDistribution"
30 pairs

"DiphtFin" "DiphtFin-s" 31.8
"DiphtFin" "DiphtFin-jen" 68.2
"DiphtPen" "DiphtPen-s" 52.7
"DiphtPen" "DiphtPen-jen" 47.3
"DiphtAntepen" "DiphtAntepen-s" 36.8
"DiphtAntepen" "DiphtAntepen-jen" 63.2
"ObsFin" "ObsFin-s" 0
"ObsFin" "ObsFin-en" 100
"ObsPen" "ObsPen-s" 17.4
"ObsPen" "ObsPen-en" 82.6
"ObsAntepen" "ObsAntepen-s" 17.1
"ObsAntepen" "ObsAntepen-en" 82.9
"SonFin" "SonFin-s" 11.9
"SonFin" "SonFin-en" 88.1
"SonPen" "SonPen-s" 72.2
"SonPen" "SonPen-en" 27.8
"SonAntepen" "SonAntepen-s" 43.7
"SonAntepen" "SonAntepen-en" 56.3
"FrontFin" "FrontFin-s" 37.0
"FrontFin" "FrontFin-jen" 63.0
"FrontPen" "FrontPen-s" 73.9
"FrontPen" "FrontPen-jen" 26.1
"FrontAntepen" "FrontAntepen-s" 55.2
"FrontAntepen" "FrontAntepen-jen" 44.8
"BackFin" "BackFin-s" 62.8
"BackFin" "BackFin-jen" 37.2
"BackPen" "BackPen-s" 85.7
"BackPen" "BackPen-jen" 14.3
"BackAntepen" "BackAntepen-s" 70.0
"BackAntepen" "BackAntepen-jen" 30.0
```