# Penalty Logic and Genomic Encoding

**MSc Thesis** (*Afstudeerscriptie*)

written by

**Teresita Mijangos Martínez**
(born April 10, 1974 in Tapachula, Chiapas; México)

under the supervision of **Dr. Reinhard Blutner**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

**MSc in Logic**

at the *Universiteit van Amsterdam.*

| **Date of the public defense:** | **Members of the Thesis Committee:** |
|---|---|
| *September 26, 2005* | Prof. dr. Peter van Emde Boas |
| | Prof. dr. Dick de Jongh |
| | Dr. Henk Zeevat |
| | Dr. Reinhard Blutner |

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

## Abstract

Universal Grammar (UG) is characterized by the assumption that all human languages share a common structure with respect to their linguistic well-formedness conditions. In Optimality Theory (OT) this structure is represented by a set of linguistic candidate forms, a set of constraints and the definition of *being optimal*. The nativist view of OT assumes that such a structure is encoded in a Language Acquisition Device (LAD) provided to a learner of a language genetically.

The symbolic and the biological levels in which OT and the LAD are respectively, can be complemented with an intermediate level: the connectionist level. In this thesis, our purpose is focus on the link of the symbolic and the connectionist level with a logical stage. Through an example based on the CV Syllable Theory of OT, we build a bridge among OT-Connectionism-Penalty Logic. In particular, for the link OT-Connectionism, we take from Smolensky and Legendre (2005) a translation of the CV Syllable Theory into connectionist terms. Such a translation is called $CV_{net}$. For the link Connectionism-Penalty Logic, we propose a translation, expressed in the penalty knowledge base *cv*. This last translation has as purpose to simplify the encoding of the common structure $\mathcal{U}$ suggested by $CV_{net}$. In fact, such a translation could be seen as an alternative encoding of $\mathcal{U}$. As advantages, it has its simplicity and the enrichment of the linguistic exploration with logic's tools.

# Acknowledgments

I gratefully acknowledge the Consejo Nacional de Ciencia y Tecnología (CONACYT) in Mexico for the scholarship this institution granted me and thereby made it possible that I studied at the ILLC. [1] In particular, I want to express my gratitute to the staff of scholarschips abroad, not only for their efficiency but also for the gentle treatment that they always gave me.

Many thanks to my advisor, Dr. Reinhard Blutner for his guidance and support through all the thesis work. His academical comments and his patience made easier the way and helped me significantly in improving the work. 'Thanks *professor*, it has been a pleasure to work with you!'.

I would like to express also my gratitute to my mentor, Professor Dick de Jongh for all the help that I received from him before and during the master. Here I only make the remark that I was fortunate in having him as mentor.

Thanks also to Professor van Emde Boas and Dr. Zeevat for their kind disposition of forming part of my Thesis Committee.

My gratitute also to Atocha Aliseda. Her advice and her unconditional academical support before I started the master were quite helpful for finishing the program.

I would like to say 'thank you' also to Ariel Campirán and Maricarmen Martínez for their comments about the thesis. I appreciate a lot that they have accepted to read my work and give me helpul insights to continue my research in a near future.

Many thanks also to my friends Sophia Katrenko, Fenrong Liu and Kasia Pilch. Their talks sometimes about the thesis, sometimes about other topics, made me enjoy the life in Amsterdam. In particular, I want to thank Sophijka for her technical advice about LATEX.

Finally, my eternal gratitude to my family and my boyfriend. 'How can I ever express my thanks to you for all you have done for me? At least, permit me to dedicate this thesis to you'.

[1] Reference. 24-month scholarship from September 2003 to August 2005. Scholar: 178224/Scholarship: 192458

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 This is the best

According to Leibniz 'this is the best of all possible worlds'. Do you believe it? Why not? Being the *best* does not mean being perfect. It means that among a set of competitors, the candidate that is chosen as the *best* one is the candidate that fulfills to a degree certain constraints. But notice that the *best* candidate is always the *best* with respect to a certain group of candidates and with respect to the fulfillment of a certain set of constraints. Optimality theory is based on this general idea. It always looks for the *optimal* or *best* linguistic form, not for the perfect one.

Optimality Theory (OT) is a framework for generating linguistic theories. It was officially presented to the intellectual community in the decade of the 1990s. From this time it has brought new and interesting insights in different areas of linguistics, especially in the area of phonology. A central idea in OT is that the knowledge of linguistic well-formedness that is attributed to speakers is universal. That does not mean that the speaker always performs linguistic forms in a well-formed manner. What it means is that the knowledge of certain structures is shared by all humans. This assumption of a core of universals in human languages identifies OT as a Universal Grammar (UG).

It is important to note that there is no consensus among UG frameworks, about which elements form the core of universals assumed for all possible human languages. Their common link is just the general assumption of the existence of such 'universals'. Furthermore, what 'universal' means in each of the UG proposals is not homogeneous. OT for example, introduces a break with the notion of 'universal' in Generative Grammar. In OT inviolability is not necessary for universality while in traditional generative frameworks it is.

A UG assumes that certain elements of human language do not vary, these are the universals. Moreover, a UG has to explain the variable elements of human languages, why they differ with respect to each other. With respect to this, OT says that language variation is a question of constraint ranking variation. In other words, certain sets of constraints are the same in all languages, but languages differ just in that each language orders such constraints in different ways. To learn a grammar is not to learn new constraints, it is only to learn another ordering of the same universal constraints.

As we have said, OT shares with other UG approaches the belief that humans share some linguistic knowledge. That this knowledge is innate is not a necessary assumption in OT. But in this thesis we want to focus on the nativist position of OT. We think that this exploration can bring us new ideas for a possible encoding of a linguistic framework.

## 1.2 The nature *vs.* nurture debate

The "nature *vs.* nurture" debate is one of the most traditional and lively debates concerning the origin of human knowledge. Unfortunately this debate has arrived at a stalemate, where the arguments given for supporting one of the positions do not seem better than those proposed for defending the other position. Here we are going to draw a general picture of the nature *vs.* nurture debate, in terms of the nativism-no nativism debate.

It is difficult to present a clear view of the debate of nativism against no nativism, mainly for two reasons. First, there are a lot of issues mixed when this debate is presented. The link nativism-no nativism is successor of the discussion rationalim-empiricism. In such a tradition many elements enter in the discussion and usually they are not cleary distinguished. Second, the mixture of issues has lead to several misunderstandings between the defenders of nativism and those who reject it.

How can we explain the nativism-no nativism debate? Traditionally nativism has been related to rationalism and no-nativism to empiricism. Nonetheless, these links can be dissolved. The point of discussion among learning theorists is not whether there is something innate or not, in the sense of something being possessed at birth before sensory experience or not. Innateness in this sense of something possessed at birth is quite accepted. The disagreement arises in the opinion about *what* is considered to be native. Following [2], we could say that the nativist controversy should be constructed in terms of *what is native* or innate not in terms of whether anything is innate.

Chomsky's strong nativism marks a clear border in the discussion. On the one side, those who agree about the genomic encoding of the abstract grammatical knowledge; on the other, those who do not. The interesting point is that connectionism —as we will see in the next section— can bring a new approach to this discussion, the approach of an integrationist view.

The nativism in grammatical knowledge compatible with OT is based on three elements: language, typology and a language acquisition device. Below the general definition:

> **The nativist hypothesis (general formulation)**
> a. *Language.* A language $\mathcal{L}$ is a rich combinatorial system governed by complex structural well-formedness conditions, $G_{\mathcal{L}}$.
> b. *Typology.* The well-formedness conditions $G_{\mathcal{L}}$ for all human languages $\mathcal{L}$ share a great deal of common ('universal') structure $\mathcal{U}$. The space of possible human languages, $\mathcal{T}$, is intrinsically structured, the boundary separating possible from impossible languages being subtly determined by the universals of well-formedness $\mathcal{U}$.
> c. *LAD.* The genome equips the learner with a Language Acquisition Device (LAD) which encodes knowledge of $\mathcal{U}$ in the following sense: the LAD is incapable of learning an impossible human language —one that violates $\mathcal{U}$— but it is capable of effectively and efficiently homing in on any target possible human language $\mathcal{L}$ in $\mathcal{T}$, given examples of well-formed items in $\mathcal{L}$ ([25]: 971-2). [1]

In OT the structure $\mathcal{U}$ shared by all possible human languages is defined by the set of output candidates produced by *Gen*, the set of well-formedness constraints *Con* and the definition of *being optimal*.[2] The learner is equipped with a Language Acquisition Device (LAD) that encodes the knowledge of the structure $\mathcal{U}$. According to the definition above, such LAD is provided by nature, is something that humans would have thanks to their genome.

## 1.3 Other traditional debate

The traditional polemic of rationalism vs. empirism clearly alive in the seventeenth century is in some sense revived in the twentieth century under the name of the nativist-no nativist polemic.

The point at issue in the rationalism vs. empiricism debate was raised in the context of epistemology. The discussion focused on the origin of human knowledge. The debate can be seen as answering the question *Which is the main source of our knowledge?* Empiricists thought that our sensory experience was the main source of our knowledge while rationalists disagreed

[1] A no nativist position will tend to postulate the least number of innate elements. For example, those who support a no nativist position could accept *a* and *b* of the definition above but it would reject *c*. That is, they could agree with the idea of a common structure $\mathcal{U}$ but disagree with the idea that such a structure has been encoded genetically.

[2] Roughly speaking, to be an optimal candidate means to be the best satisfying a hierarchy of constraints.

about it and thought that ‘true knowledge’ was based on the principles of reasoning.

Empiricists drew the distinction between simple and complex ideas. Simple ideas were those ideas directly produced from sensory experience. Complex ideas were composed from simple ideas. The empiricist tradition includes an account for psychological processes called *associationism.* In this view complex ideas are thought of as associations of simple ideas. The association can be made for example by spatial contiguity, or by temporal contiguity. The point that we want to stress here is that for empiricists, ideas were grounded in experience in the sense that every idea is related directly or indirectly to it. Rationalists do not think so. For them ideas are innate, inborn, not acquired through sensory experience. True beliefs arrive by reasoning according to these ideas.

The rationalist-empiricist debate can be brought into the context of language acquisition. The debate was revived in the middle of the $20^{th}$ century by Chomsky. In his polemic argument about the poverty of stimulus he stated the impossibility of an organism of learning everything from experience. The debate was nativism-no nativism, exemplified under the names Chomsky against Skinner.

Behaviorism, mainly represented by Skinner, is a kind of associationism. Universal Grammar ocurring as generative grammar in Chomsky, is innatism. The routes seem to be empiricism-associationism-behaviorism-Skinner and rationalism-Universal Grammar-generative grammar-Chomsky. The context in which connectionism[3] appears it assumes these links. Connectionism ‘is an *elaboration* of associationism’ (*Cfr.* [2]). So, it is not strange that the first reaction of those who supported nativism was rejection. Even when connectionism has incorporated ideas of associationism, it has added new ideas that places it far from previous forms of associationism. It is important to note that connectionism can be seen as neither supporting nativism nor supporting no nativism. It can be seen as a theory integrating both positions. In this tone in [24] it is said that *Parallel Distributed Processing* models can bring us a new perspective to think about the nativism-empiricism debate.

> [With respect to an organism that consists of highly interconnected units]
> Like good nativists, we have given the organism a starting point that has been selected by its evolutionary history. We have not, however, strapped the organism with the rigid predeterminism that traditionally goes along with the nativist view (...) At the same time, we have the best of the empiricist view, namely, we place no *a priori* limitations on how the organism may adapt to

[3] By connectionism here we mainly mean an integrative connectionism. We exclude for example, the eliminativist and implementationist positions.

> its enviroment. We do, however, throw out the weakest aspect of the empiricist dogma, namely, the idea of *tabula rasa* (or totally random net) as starting point ([24]: 140-1).

We close this section with an idea that other authors have mentioned before in [2],[11], [24]: Connectionism may be an option for closing the gap in the nativism-no nativism debate.

## 1.4 What is innate?

> The answer is not Nature *or* Nurture; it's Nature *and* Nurture. But to say that is to trade one platitude for another; what is necessary is to understand the nature of that interaction ([11]:357).

The concept of innateness has a quite wide use in different areas. Here we are interested in a neural sense of the term. This does not mean a homogeneous definition of the term. Innateness as connected with biology, brings us the general idea of genes and different characterizations of innateness that in many cases are not clear. For example, *being innate* from the biological perspective can refer to our complete genetic endowment, or only to a single set $g$ of genes that produces a specific behavior, and so on.

In [11], the authors argue that for a precise characterization of innateness it is important to distinguish between the *mechanism* and the *content* of innateness. We focus only on the *mechanism*, given its relevance for our topic.

With respect to the *mechanism*, we can identify three kind of constraints: representational, architectural and timing constraints.

*Representational* constraints in a network level constitute patterns of activation that spread across the processing units. Activation patterns are determined considering connections between units. Innateness may be explained in terms of *prespecified* weights on the connections.

*Architectural* constraints operate in three levels: unit, local and global level. But it is important to stress that in all levels, such constraints operate considering the *architecture* of a network.

In the unit level, the focus is on the nodes of the network. In this level, innateness is explained in terms of activation function, learning algorithm, and so on. In the local level, the focus is on patterns of connectivity, for example, it would take into account whether the network is feedforward or recurrent. In this level innateness is related to network type, number of layers, of units, and so on. In the global level, the focus is on the macro view, that is, in the view that different pieces of a system form a unity. In this level, all elements of the system must be taken into account for explaining certain phenomenon. Innateness would be understood as emerging from the relation of different parts of a network.

*Timing* constraints arise from the timing of developmental events. Here, innateness is understood in terms of adaptative learning rates, incremental presentation of data, and so on.

It is worthwhile to mention that only in the representational level a neural mechanism is capable of dealing with the idea that knowledge of grammar can be innately specified. UG and, specifically OT, will be related to the representational level. So, innate knowledge in the OT framework will take the form of prespecified weights on the unit connections.

Universal Grammar assumes that certain grammatical knowledge is innate. However, as Elman has pointed out in [11], representational constraints even when theoretically plausible seem to be biologically implausible. This point reminds us of the topic of *competence* and *performance.* A grammar is oriented to explain the *competence* of language not its performance. That is, a grammar is focused on explaining the *use* of the language by a speaker as well as the judgment of grammaticality. Nonetheless, it does not deal with the issue of how a speaker processes the language. This is an important issue to be noted in order to understand why for a proposal in linguistics, as OT for example, to use the representational level on constraints is relevant even when such constraints are far from being a biologically plausible model of biological networks.

A last remark with respect to *the translation invariance problem.* [4] For solving *the translation invariance problem*, OT proposes to bound weights. This proposal is in tone with the proposal of [12]. Proposals that suggest as solution of *the translation invariance problem* to make the weights identical, usually assume representational constraints. According to what we have mentioned before, this would mean that networks built under the assumption of such proposals cannot be seen as plausible biological models, as explanation of *how* speakers process language. Nevertheless, such networks perhaps are not far from explaining how speakers *use* the language. [5]

## 1.5 The proposal

In this thesis we propose to translate the constraints of the symmetric network $CV_{net}$ into formulae of penalty logic. The novelty of this proposal is in the link penalty logic-optimality theory-connectionist networks, expressed specifically in the link penalty logic-$CV_{net}$. The proposal is located in an interdisciplinary environment. Some areas in which it could generate new issues of research would be logic, linguistics, computation and philosophy.

[4] For an introduction to the translation invariance problem, *Cfr.* [12].

[5] This point is important for us given that we will translate a network in which bound weights and representational constraints are assumed.

## 1.6 About the other chapters

**2 Optimality Theory**

OT assumes that there are three things that all human languages share: (i)A set of linguistic forms, from which the *best* or *optimal* form is selected; (ii) a set of constraints and (iii) the definition of *being optimal.* These three elements constitute the common structure $\mathcal{U}$. The genome brings to the learner a Language Acquisition Device (LAD) in which $\mathcal{U}$ is encoded.

In this chapter we introduce the elements of $\mathcal{U}$. We present a concrete example that implements such a structure, the CV Syllable Theory.

**3 CV$_{net}$: A network embodying OT**

In this chapter we treat the encoding of $\mathcal{U}$ into LAD. In particular, we look at the encoding of the CV theory into a network called $CV_{net}$. This network is part of the LAD.

**4 Penalty Logic**

The genomic encoding of $\mathcal{U}$ into the connectionist network $CV_{net}$ can be simplified and enriched if we encode $\mathcal{U}$ in Penalty Logic.

In this chapter we introduce the main definitions of Penalty Logic and we exemplify how a symmetric network can be translated into Penalty Logic.

**5 Translation of $CV_{net}$ into Penalty Logic**

In this chapter we present the genomic encoding of OT into Penalty Logic. The $CV_{net}$ network is translated into Penalty Logic. The link among OT-Connectionist-Penalty Logic's levels is closed.

# Chapter 2

# Optimality Theory

In this chapter we present some elements of the framework of Optimality Theory (OT). As we have pointed out in the introduction, we assume a nativist reading of OT. The reason for such assumption is only methodological. We think that the exploration of the nativist path might bring us new ideas for a more critical view of the nature *vs.* nurture debate [1].

We start this chapter, defining a 'nativist' OT. The definition will permit us to present a rough introduction of some elements of OT as well as some clarifications about the set of universals in an OT Theory. After the nativist OT's definition, we present more extensively some elements of OT. We close this chapter with the presentation of the CV Syllable Theory. The CV Theory will be the base for the examples developed in later chapters.

## 2.1 The nativist hypothesis in OT

According to [25], the nativism in OT can be defined as follows:

> **Definition 2.1 The nativist hypothesis in OT**
> a. *Language.* A language $\mathcal{L}$ is a subset of a combinatorially-structured candidate set of linguistic forms $\mathcal{S}$; $\mathcal{L}$ consists of those forms which best satisfy a set of violable well-formedness constraints $Con_{\mathcal{L}}$, as ranked in a strict domination hierarchy, $G_{\mathcal{L}}$.
> b. *Typology.* The set of candidate forms $\mathcal{S}$, the set of well formedness conditions *Con*, and the formal definition of 'best satisfying a hierarchy of violable constraints' are exactly the same for all possible human languages: they define $\mathcal{U}$. The typological space of possible human languages $\mathcal{T}$ is the space of all languages $\mathcal{L}$ comprised of the optimal forms relative to some rankings $G_{\mathcal{L}}$

[1] In fact, we think that exploring only the nativist side is not enough for a critical view of the debate. A research in the no nativist side is also required for a mature valuation of both positions.

of the constraints *Con*.
c. *LAD*. The genome $g$ equips the learner with a Language Acquisition Device (LAD) encoding knowledge of $\mathcal{U}$ in the following sense.

i. During development, $g$ drives the construction of a neural network $\mathcal{N}$ that maximizes Harmony.
ii. The activation patterns in $\mathcal{N}$ can realize any candidate linguistic structure in $\mathcal{S}$.
iii. The connections in $\mathcal{N}$ encode the constraints *Con*: maximizing Harmony constructs the realization of the structure that best-satisfies the constraints *Con*, with each constraint $\mathcal{C}_i$ in *Con* having a numerical strength $s_i$ .
iv. During language acquisition, $\mathcal{N}$ adjusts the constraint strengths $s_i$ while processing example linguistic forms in $\mathcal{S}$. The learning algorithm finds a set of strengths $s_i$ such that each constraint in *Con* strictly dominates all weaker constraints —i.e., it finds the grammar of a possible language in $\mathcal{T}$.
v. If the learning data derive from a language $\mathcal{L}$ in $\mathcal{T}$, the learning algorithm efficiently converges to a set of strenghts $s_i$ realizing a ranking of *Con* that correctly generates $\mathcal{L}$. ([25]: 972).

OT deals with linguistic forms. From the set of linguistic forms, some forms arise as *optimal*. The selection of an *optimal* form is not absolute, but relative to a certain hierarchy of the constraints *Con*. The set of *optimal* forms that emerge from a candidate evaluation in a hierarchy, say H, constitute a language.

In OT, a grammatical form is an *optimal* form. In order to select such a form, there is a candidate comparison, in which, the constraint hierarchy is the most important tool for deciding which form is the *optimal*. It is important to note, that for OT a grammatical form *is not* a form that does not violate any constraint. On the contrary, *optimal* forms usually violate some constraints. An important point for selecting the *optimal* form is the position of the violated constraint(s) in a certain hierarchy. Constraints at the top of the hierarchy have a higher value. As the hierarchy goes down, the value of constraints decrease. Violations on the bottom constraints are preferable given their lower cost in comparison with the top constraints. However, the position of the violated constraint(s) is not sufficient for selecting the winner candidate. For selecting the *best* form it should be also taken in account which constraints the other competitors have violated.

In definition 2.1, specifically in the paragraph about *typology*, two important statements about OT are introduced. First, it is said what is going to constitute the common structure for all possible human languages, that is, the structure $\mathcal{U}$. This common base that all human languages share is formed by the set of linguistic forms,[2] the set of violable constraints *Con* and the definition of 'being optimal'. That means that human languages do not differ neither in the candidate forms to be consider in the selection of *optimal* forms, nor in the constraints that regulate the selection, nor in what they consider as an 'optimal' form. According to OT, languages only differ in the ranking of constraints. Second. "Typology is the study of the range of systems that re-ranking permits" ([20]:7). According to the definition above, the typological space of possible human languages is constituted by the space of all languages $\mathcal{L}$. That is, by all the *optimal* forms that arise through different re-rankings of the constraints in *Con*.

Definition 2.1 also tells us that the structure $\mathcal{U}$ is encoded in a Language Acquisition Device(LAD). This LAD is a harmonic network call it $CV_{net}$. The point that I want to stress here is that the genome provides such LAD. The genomic origin of the LAD has led people to see $CV_{net}$ as an inborn device. Such assumption about $CV_{net}$ makes it plausible to think that the definition of 'being optimal', the linguistic forms and the set of constraints *Con*, due to their encoding in $CV_{net}$, are inborn too.

A more complete characterization of the inborn character of $\mathcal{U}$, specifically in $CV_{net}$, should consider not only the violable constraints but also the inviolable. In the CV Syllable Theory as formulated in ([25]: 980-1), the distinction between violable and inviolable constraints is explicitly drawn. Violable constraints are the *Con* constraints. Inviolable constraints are the *Gen* and the *Structural* constraints. As the authors state in [25], the *Gen* and the *Structural* constraints are beyond any constraint hierarchy, they are assumed fulfilled in every candidate competition. As you can see in the definition 2.1, such constraints are not mentioned. This suggests us the alternative that such inviolable constraints may be excluded from the CV Theory. However, independently whether a set of *Gen* and *Structural* constraints belong or not to the general OT, what we want to stress here is that we are going to consider such constraints as part of the structure $\mathcal{U}$ that is encoded in $CV_{net}$. We think that we are allowed to do this, first, because such constraints as presented in [25] seem to be more fundamental than any violable constraint; second, they are part of $CV_{net}$.

[2] Note that it is the whole set of linguistic forms, this includes not only *optimal* but also *no optimal* forms.

## 2.2 The OT framework

Optimality Theory arises to the public light at the end of the $20^{th}$ century. Alan Prince and Paul Smolensky introduced this framework extensively in [20]. Initially, OT was focused on phonology, but since its birth has been promptly extended to other areas of linguistics like syntax and semantics.

OT emerges in a context in which linguistic research is mainly based on the framework of generative grammar.[3] However, it differs from the traditional approach in important points:

- In OT, linguistic forms are compared to each other for the selection of an optimal. The status of a form always depends on other forms. In generative grammar, this is not the case. Forms are singly evaluated according to certain rules.
- OT's constraints are universal and violable. Generative grammar's principles are universal and inviolable. [4]
- OT recuperates *markedness* as a key element of a grammar where in generative grammar the central role was played only by faithfulness. This recuperation is related to the conflicting relation that the constraints will hold in OT.
- In OT, neither is violability a synomyn of inactivity, nor does satisfaction (inviolability) mean activation. The OT outputs are not thought in terms of duality, e.g. on/off, right/wrong, and so on, but in terms of optimization. In generative grammar violability is inactivation, satisfaction means activation.
- OT explains language variation in terms of re-ranking of constraints, while generative grammar justifies that aspect appealing to other tools like different rules/parameters and the lexicon.
- In generative grammar the lexicon plays an important role imposing structure on the linguistic primitives. In OT, linguistic primitives and

[3] In the early 80's Chomsky presents the Principles and Parameters Theory (P& P) (*Cfr.* [7] and [8]). In P & P, it is postulated that languages do not have rules but principles which are universal. Language variation is explained through parameters."The reliance on principles rather than rules has consequences also for the interpretation of the term **generative grammar** that has been been associated with the Chomskyan approach since it first appeared. 'Generative' means that the description is rigorous and explicit; 'when we speak of the linguistic's grammar as a 'generative grammar' we mean only that it is sufficiently explicit to determine how sentences of the language are in fact characterized by the grammar ([6]:220)'". ([10]:35).

[4] Acording to the CV Syllable Theory as exposed in ([25]: 980-1), some constraints are universal and violable, these are the *Con* constraints. The *Gen* and the *Structural* constraints are universal but inviolable. This is a distinction with respect to generative grammar in which all the principles are universal and inviolable.

inputs are the same (*Richness of the Base Hypothesis*). From this point of view, two positions about the lexicon arise in OT: *a*)If a lexicon is recognized, the role of the lexicon is quite different and almost null; or *b*) There is no lexicon, only vocabulary.

- In OT and generative grammar, Universal Grammar (UG) plays a central role. But the conception of UG that underlies these frameworks is quite different. In OT, UG consists mainly of a set of universal and violable constraints, while in generative grammar UG is a set of (inviolable) principles and rule schemata.

It will be helpful for the clarity of the next sections, to mention something about the relationship between OT and OT grammars. OT constitutes a theory of human language capacity while an OT grammar is a part of that language capacity. OT is a framework for positing linguistic theories, its instantiations are particular OT grammars or OT linguistic theories. OT deals with the structure of UG, while the grammars are concerned with its content. In other words, OT works with the typology, with all possible re-rankings of the set of universal constraints *Con*; OT grammars are the particular rankings in that space of possible re-rankings.

## 2.3 The Basic Architecture

An OT grammar can be seen as an input-output mechanism[5]. The grammar must assign to each input or underlying form, an output or surface form. This input-output pairing is such that to each input there corresponds only one output, and this output is always the *optimal* form among a set of competitors.

The input-output association is accomplished by two functions: the Generator (hereafter *Gen*) and the Evaluator (henceforth Eval). *Gen* associates with each input a set of possible analyses or output candidates. This set is submitted to Eval. Eval will evaluate the candidate output forms using a ranking of the constraints in *Con*. From this evaluation, the *optimal* candidate will be selected as the output.

The process described above can be schematically represented like:

$$\text{Gen}(I_k) \rightarrow \{\ c_1,\ c_2,\ldots,\ c_n\ \}$$
$$\text{H-Eval}\ (c_i,\ 1 \leq i \leq \infty) \rightarrow \text{Out}\ ^6.$$

[5] In this section, the description of the mechanism for generating an optimal form is based on [20]. In the section 'Structural Descriptions' we complete this exposition mentioning some elements of the contemporary OT as exposed in [25].

[6] Prince and Smolensky describe this process in similar terms in [20]. Here we have changed only the notation.

where $I_k$ is the input for which a set of candidates $c_1$, $c_2$,..., $c_n$ will be generated. As we have said Eval selects from that set of candidates, a candidate $c_i$ that becomes the output (Out) assigned to the input $I_k$. In such selection a ranking or constraint hierarchy H is assumed.

The number of candidates that *Gen* can assign to a certain input $I$ can be infinite. Besides, *Gen* is free to generate any conceivable candidate for a certain input *I*, even when such output candidate may be a quite unfaithful analysis of that input. The unique requirement is that *Gen* uses licit elements for building the candidates, that is, elements from the universal vocabulary or lexicon.

*Gen*'s characteristic of postulating any amount of analyses or output candidates for a certain input *I*, is known as *Freedom of Analysis.* This freedom of *Gen* could raise the question whether the system can at some point finish the evaluation and state an output. This question leads us to the distinction, usually accepted among formal linguists, between the grammar and its cognitive implementation. That is, the distinction between *competence* and *performance.* A grammar model is satisfactory if it explains grammatical judments of speakers and regularities in natural language. Its aim is not to explain how human minds process linguistic knowledge but how they use them. In other words, a grammar's goal is oriented to the competence of a language not to its performance.[7] The problematic issue of whether a system can finish the process of evaluation and to select an output is a problem related to the performance of language not to its competence. So, it seems to be a problem that goes beyond a grammar's goal.[8]

The candidates generated by *Gen* are compared to each other. From this comparison, Eval will choose an optimal candidate, that is, the candidate that best satisfies the ranking of the constraints. Independently of the number of output candidates, Eval always selects one, the most *harmonic* candidate that becomes the output. There are two cases that may seem problematic with respect to the unicity of the output. First, when there are two or more candidates that have the same number of violation marks with respect to the same constraint. In this case OT will consider such candidates as forms equally correct. Second, when there is no optimal candidate. OT always assumes that the candidates generated by *Gen* can be compared in terms of violation marks. The candidates have the *same* number of marks[9]

[7] It leaves to disciplines like psycholinguistics, neurolinguistics and computational linguistics, the task to deal with the cognitive implementation or performance. In Chomsky, the distinction between competence and performance can be drawn in terms of the distinction between I-language and E-language ([5]).

[8] It should be observed that the gap between competence and performance is been closed. Perhaps this explains why more people from 'theoretical' linguistics, computational linguistics and neurolinguistics are nowadays working together.

[9] Having the *same* number of marks means in an OT context, the candidates compared with respect to a certain constraint $\mathcal{C}$ in a hierarchy H, have the same *number* of violation marks. Violation marks are usually represented in OT by asterisks.

or not. If they do not, there is always a candidate with the minimum number of marks.

Summing up. The basic architecture of an OT grammar includes a universal candidate generator *Gen*, a universal evaluation function Eval, and a universal set of constraints *Con*, whose constraints are ranked in a hierarchy H. Assumming that a grammar deals with competence not with performance, Eval is always able to find an output for a given input.

## 2.4 Two readings of an OT architecture

The basic architecture of an OT grammar introduced in the previous section can be depicted as in figure 2.1. This architecture can be read in terms of *harmonic serialism* or in terms of *parallelism.* In *harmonic serialism* the production of an *optimal* form is explained as follows: A set of candidates generated by *Gen* is advanced to evaluation, where Eval assuming a hierarchy of the constraints selects the most *harmonic.* The *optimal* candidate is fed back into *Gen. Gen* generates another set of candidates, which again is evaluated, and again the *optimal* candidate is returned to *Gen.* This process continues until no improvement in optimization is possible [20].

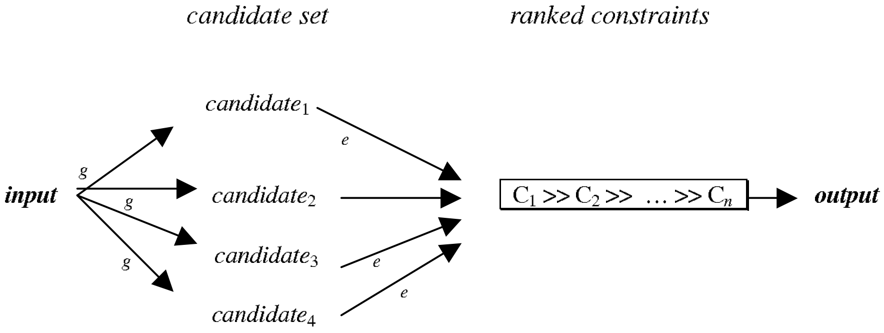


Figure 2.1: Basic OT Architecture

The other interpretation called *parallelism* assumes that *Gen* generates all the candidates at once. The candidate evaluation is *global* and *parallel.* Just one pass through the constraint hierarchy is sufficient for selecting the most *harmonic* candidate, that is, the output. In this thesis, we will assume this second approach.

## 2.5 The optimal candidate

Intuitively, the *optimal* candidate that constitutes the output of an evaluation, it is the winner among other candidates. In the selection of an *optimal* candidate, the number of violations on the constraints is not as important as the position of the constraint(s) in a certain hierarchy. To violate a higher constraint is more costly than to violate a lower one. An *optimal* candidate is always optimal with respect to other candidates. The *best* or *optimal* form will be that candidate that violates lower ranked constraints than those constraints that the other candidates violate. That is,

> **Definition 2.2**
> *A candidate w is considered to be optimal iff for each competitor w', the constraints that are violated by w must be ranked lower than at least one constraint violated by w'.*(*Cfr.*,[3]:20).

## 2.6 Speakers and Hearers

As you can see in figure 2.1, we have depicted a forward relation between input and output, but the relation can be backward. What we have called input and output are respectively the underlying form and the *optimal* form. Figure 2.1 shows us the relation from the underlying form to the *optimal* form, this relation is a function that we call, following [25], 'the production function'. This function denoted by $f_{prod}$, can be identified with the *speaker* perspective.

Additionally, in [25] a backward function is introduced. This function maps from the *optimal* candidate[10] to the underlying form, called it 'the comprehension function', denoted by $f_{comp}$, and it is commonly matched to the *hearer* perspective.

In this thesis we focus on the speaker perspective. We will use the terms 'input' and 'output' as in earlier sections, as denoting the underlying and the *optimal* forms respectively. A more precise characterization of the functions just exposed is given in the next section, with the help of the concept of structural description.

[10] With more precision, the function maps from the *speaker's optimal candidate* to the underlying form. It is important to note this, given that the speaker's and hearer's optimal candidates may not coincide. In this thesis, we use the term 'optimal' only in the speaker's perspective.

## 2.7 Structural Descriptions

[11] Structural descriptions play an important role in theories that assumes the principle of compositionality of meaning. This principle states that the meaning of a complex expression is formed from the meaning of its parts plus rules for combining such parts. OT grammars assume the principle of compositionality. Structural descriptions will be the parts from which compound structures will be formed.

A structural description in OT has three primitive elements: an interpretation, an expression and a correspondence relation between the interpretation and the expression. In the speaker's perspective, input elements are interpretations while outputs elements are expressions. [12] For example, in the Basic CV Syllable Theory of OT, interpretations are strings of consonants and vowels and expressions are such strings parsed into syllables. [13]

**Definition 2.3**
A structural description $\mathcal{S}$ in an OT grammar is a tuple ($I$, $E$, $\mathcal{R}$), where

1. $I$ is an interpretation
2. $E$ is an expression.
3. $\mathcal{R}$ is a correspondence relation between $I$ and $E$.

For every expression $E$, there is a substructure $O$, called the *overt part.* [14] The universe of structural descriptions is denoted by $U_{\mathcal{S}}$. The universe of interpretations, expressions, correspondence relations and overt forms by $U_I$, $U_E$, $U_{\mathcal{R}}$ and $U_O$ respectively.[15]

The set of candidate expressions of a particular interpretation $I_0$ is the set of all structural descriptions $\mathcal{S} = (I_0, E, \mathcal{R})$ in $U_{\mathcal{S}}$. Such a set is called $Gen(I_0)$. That is,

[11] The content of this section is based on [25]:472-495.

[12] In the hearer's perspective is in the opposite way. Input elements are expressions while outputs are interpretations.

[13] We can take the string /V$^1C^2$ $V^3C^4$/ as interpretation. Its corresponding *grammatical* or *optimal* expression with respect to the hierarchy *Fill*$^C$ $\gg$ *Parse* $\gg$ *Fill*$^V$ $\gg$ *NoCoda* $\gg$ *Onset* would be .V$^1$.C$^2$V$^3$C$^4$. *Cfr.* section 2.9

[14] The *overt part* or *surface form* is 'the portion of $E$ that is directly available to the hearer'([25]: 472).

[15] Each OT grammar specifies a universe of structural description, that is, a $U_{\mathcal{S}}$.

**Definition 2.4**

$$Gen(I_0) = \{\ \mathcal{S} \in U_{\mathcal{S}} |\ \exists E \in U_E, \mathcal{R} \in U_{\mathcal{R}} \text{ s.t. } \mathcal{S}= (I_0,\ E,\ \mathcal{R})\}$$

The so-called function *Gen* is the correspondence relation that maps from the interpretation $I_0$ to the *Es* $\in U_E$. The *production function* assigns to each interpretation a subset of the set assigned by *Gen.* That is, $f_{prod}(I_0) \subset Gen(I_0)$. If a structure $\mathcal{S}= (I_0,\ E,\ \mathcal{R}) \in f_{prod}$, $\mathcal{S}$ is a grammatical expression of the interpretation $I_0$.

The set of candidate interpretations of a particular expression $E_0$ is the set of all structural descriptions $\mathcal{S} = (I, E_0,\ \mathcal{R})$ in $U_{\mathcal{S}}$. This set is called $Int(E_0)$.

**Definition 2.5**

$$Int(E_0) = \{\ \mathcal{S} \in U_{\mathcal{S}} |\ \exists I \in U_I, \mathcal{R} \in U_{\mathcal{R}} \text{ s.t. } \mathcal{S}= (I,\ E_0,\ \mathcal{R})\}$$

The function *Int* maps from the expression $E_0$ to the *Is* $\in U_I$. The *comprehension function* assigns to each expression a subset of the set assigned by *Int.* That is, $f_{comp}(E_0) \subset Int(E_0)$. If a structure $\mathcal{S}= (I,\ E_0,\ \mathcal{R}) \in f_{comp}$, $\mathcal{S}$ is a grammatical interpretation of the expression $E_0$.

As we have mentioned, we will focus on the speaker's perspective. We consider the hearer's perspective also interesting , but it goes beyond our main goal in this thesis.

## 2.8 The constraint hierarchy

The function of evaluation, that is, Eval, carries the responsability of selecting the grammatical well-formed structures in a certain language. Assuming a strict order on the constraints, Eval looks for the output in the space of candidates produced by *Gen.* But for such an evaluation gets help from the following devices: a constraints hierarchy, marking of violations and harmony evaluation ([15]). In this section we only comment the first device.

In OT, the universal constraints grouped in *Con*, are not mutually consistent. In order to resolve their conflict and to choose the most *harmonic* candidate, all the constraints in *Con* are ordered in a *strict domination hierarchy.* We will represent this domination as: $\mathcal{C}_1 \gg \mathcal{C}_2 \gg ... \ \mathcal{C}_n$ where '$\mathcal{C}_a \gg \mathcal{C}_b$' is read as 'constraint $a$ dominates constraint $b$'.

The strict character of the domination means absolute priority of one constraint over another. If we interpret violability in terms of numerical penalty, $\mathcal{C}_1 \gg \mathcal{C}_2$ could be read as: 'the penalty for violating $\mathcal{C}_1$ is much greater than the penalty for violating $\mathcal{C}_2$'.

The absolute priority of $\mathcal{C}_1$ over $\mathcal{C}_2$ consists in that independently of the number of violations of $\mathcal{C}_2$, large, small or nil, the violations of $\mathcal{C}_2$ does not compensate a single violation of $\mathcal{C}_1$. "The penalty for violating $\mathcal{C}_1$ is *infinitely* greater than the penalty for violating $\mathcal{C}_2$" ([20]:236).

The *strict domination hierarchy* can be seen as a strict partial order (SPO). An SPO is a pair $(\mathcal{W},\mathcal{R})$, where the relation $\mathcal{R}$ is irreflexive ($\forall x \neg \mathcal{R}xx$) and transitive ($\forall xyz(\mathcal{R}xy \wedge \mathcal{R}yz \rightarrow \mathcal{R}xz)$). In the case that concerns us, $W = Con$, and $\mathcal{R}$ is the relation $\gg$. The properties of irreflexivity and transitivity of the ranking of constraints can be expressed as:

*Irreflexivity*: $\neg$ $(\mathcal{C}_1 \gg \mathcal{C}_1)$
*Transitivity*: *If* $\mathcal{C}_1 \gg \mathcal{C}_2$ *and* $\mathcal{C}_2 \gg \mathcal{C}_3$ *then* $\mathcal{C}_1 \gg \mathcal{C}_3$.

Another property related to the ranking of constraints is called *economy.*

> ***Economy Property***: Banned options are available only to avoid violations on higher-ranked constraints and can only be used *minimally* ([20]:32).

This property is quite important, it tell us about the purpose of violating a constraint: a constraint may be violated only to avoid the violation of higher ranked constraints. And only to avoid such violations, that is, that violations even when permitted should be kept to a minimum.

For example, take the case in which markedness constraints dominate faithfulness constraints (markedness $\gg$ faithfulness). Assuming the economy property, those output candidates that violate faithfulness without reduction of markedness will never become optimal. This is in tone with the economy property, violation of a constraint (faithfulness in this case) is permitted in order to avoid violation of a higher ranked constraint (here, markedness). If the reverse ranking is the case, that is, faithfulness$\gg$ markedness, violations of markedness are justifiable in order to avoid violations of faithfulness.

## 2.9 The Basic CV Syllable Theory

For the presentation of the Basic Syllable Theory in OT, we follow [25]. This version of the CV theory assumes elements of the correspondence theory of McCarthy and Prince [18].

The common base that all human languages share, that is the structure $\mathcal{U}$, was introduced in definition 2.1. According to this definition, $\mathcal{U}$ was formed by a set of candidate forms $\mathcal{S}$, a set of constraints *Con* and the definition of 'being optimal'. We mentioned that besides such elements we would add to $\mathcal{U}$, in the case of the CV Theory, two sets of constraints: the *Gen* and the *Structural* constraints.

**a) The universe $U_{\mathcal{S}}$**

In the Basic CV Syllable Theory, the universe of structural descriptions $U_{\mathcal{S}}$, is formed by strings of consonants and vowels, that is, C's and V's. Interpretations are strings of C's and V's like /VCVC/, /CVCVC/, and so on. Expressions are strings of C's and V's but parsed into syllables. For example, from the interpretation /VCVC/ = $/\mathrm{V}^1 C^2\ V^3 C^4/$ the following candidates (i)-(iv) can be generated:[16]

| | /VCVC/ | $/\mathrm{V}^1 C^2\ V^3 C^4/$ |
|---|---|---|
| (i) | $.\square\mathrm{V.CV.}\langle C\rangle$ | $[_\sigma \mathrm{CV}^1][_\sigma \mathrm{C}^2\mathrm{V}^3]$ |
| (ii) | $\langle V\rangle\mathrm{.CV.}\langle C\rangle$ | $[_\sigma \mathrm{C}^2\mathrm{V}^3]$ |
| (iii) | $\langle V\rangle\mathrm{.CV.C}\square.$ | $[_\sigma \mathrm{C}^2\mathrm{V}^3][_\sigma \mathrm{C}^4\mathrm{V}]$ |
| (iv) | .V.CVC. | $[_\sigma \mathrm{V}^1][_\sigma \mathrm{C}^2\mathrm{V}^3\mathrm{C}^4]$ |

From the speaker's perspective, interpretations are always the inputs of the *Gen* function, and expressions the output of such function.

**b) Constraints**

The Basic CV Syllable Theory has the following constraints ([25]:980-1):

*Con constraints*

(i) *Onset.* Every syllable nucleus has a preceding onset.

(ii) *NoCoda.* Codas are not permitted.

(iii) *Parse.* For every element in the input there is a corresponding element in the output.

(iv) *Fill$^V$*. Every syllable nucleus in the output has a corresponding V in the input.

(v) *Fill$^C$*. Every syllable onset or coda in the output has a corresponding C in the input. [17]

[16] The example is shown in two notations. The notation used in the left expressions can be read as follows: .X. as 'the string X is a syllable', $\langle x\rangle$ as 'the element x is unparsed', □ as 'a node *Onset*, *Nucleus* or *Coda* is empty', x́ as 'the element x is a nucleus' ([20]:110). In the right expressions, all of the elements of the underlying form are numerated. This strategy makes the □ and $\langle\rangle$ symbols unnecessary. A syllable is indicated by $\sigma$ . In what follows we add to the left notation upper numbers just as to the right one. Numbers make clearer which elements are mapped. But we still preserving boxes and angles because they will help us in the next chapter to clarify which elements are parsed and which ones not.

[17] Another names given to *Parse* and *Fill* are MAX and DEP respectively. We will use the names of *Parse* and *Fill* as in [25] due to the distinction that the authors drawn there between *Fill$^V$* and *Fill$^C$*.

*Gen constraints*
(vi)*Identity.* Each correspondence index $i$ may label at most one pairing: either $C^i \leftrightarrow C^i$ or $V^i \leftrightarrow V^i$, but not both.
(vii)*Linearity.* Output segments mantain the order of their corresponding input segments.
(viii)*Integrity.* Each segment in the input corresponds to at most one segment in the output.
(ix) *Uniformity.* Each segment in the output corresponds to at most one segment in the input.

*Structural constraints*
(x) *Outputidentity.* Every output segment is either an onset, a coda or a nucleus, but no more than one of these.
(xi)*NoOutputgaps.* There are no gaps between consecutive segments in an output string.
(xii) *Correspondence.* No correspondence relation exists without both an input and an output element.
(xiii) *Nucleus.* There must be a nucleus, after every onset and before every coda.

The structure $\mathcal{U}$ in CV Theory is constituted by:

$$\mathcal{U}_{CV} = \{\ U_{\mathcal{S}},\ \textit{constraints},\ \textit{definition of `being optimal'}\ \}$$

In the following chapters we will focus on the encoding of the structure $\mathcal{U}_{CV}$. First, we will present the encoding in the connectionist level (chapter 3) and then in penalty logic (chapter 5).

**c) The OT Table**
In the Basic CV Theory the function *Gen* produces candidate expressions. These candidates are compared taking in account a specific hierarchy H of the constraints in *Con.* From such comparison, Eval will select the *optimal* expression in the hierarchy H.

Suppose that for the input /$\mathrm{V}^1 C^2\ V^3 C^4$/ *Gen* has generated the following candidates:

$c_1$: $.\square \mathrm{V}^1.\mathrm{C}^2\mathrm{V}^3.\langle C\rangle$

$c_2$: $\langle V\rangle.\mathrm{C}^2\mathrm{V}^3.\langle C\rangle$

$c_3$: $\langle V\rangle.\mathrm{C}^2\mathrm{V}^3.\mathrm{C}^4\square.$

$c_4$: $.\mathrm{V}^1.\mathrm{C}^2\mathrm{V}^3\mathrm{C}^4.$

And assume that the constraints in *Con* show the ranking: $\textit{Fill}^C \gg \textit{Parse} \gg \textit{Fill}^V \gg \textit{NoCoda} \gg \textit{Onset}$. [18]

The OT table that shows the candidate comparison can depicted as in table 2.1.

| input: $/V^1C^2V^3C^4/ \rightarrow$ | $\textit{Fill}^C \gg \textit{Parse} \gg \textit{Fill}^V \gg \textit{NoCoda} \gg \textit{Onset}$ | | | | |
|---|---|---|---|---|---|
| $c_1$: $.\square V^1.C^2V^3.\langle C\rangle$ | * | * | | | |
| $c_2$: $\langle V\rangle.C^2V^3.\langle C\rangle$ | | ** | | | |
| $c_3$: $\langle V\rangle.C^2V^3.C^4\square.$ | | * | * | | |
| $c_4$: $.V^1.C^2V^3C^4.$ | | | | * | * |

Table 2.1: Candidate comparison in OT

Asterisks indicate violations on constraints. For example, candidate $c_1$ has violated the constraints *Fill*$^C$ and *Parse*. If you observe the table, all candidates have violated a constraint. The *optimal* candidate is $c_4$ because from each competitor c', the constraints that are violated by $c_4$, that is, *NoCoda* and *Onset*, are ranked lower than at least one constraint lost by any c' (definition 2.1).

Now, suppose that we have the same input and output candidates as above, but the order that the constraint in *Con* show is different. Take the hierarchy H' being: $\textit{Onset} \gg \textit{NoCoda} \gg \textit{Fill}^V \gg \textit{Parse} \gg \textit{Fill}^C$. The corresponding OT table is table 2.2.

With respect to the constraint hierarchy H', candidate $c_1$ arises as the *optimal* form.

Different rankings on the constraints in *Con* can support different linguistic forms as *optimal*. This means that grammatical forms will be relative to the contraint hierarchy assumed. As we have mentioned, in a nativist OT, human languages will share the structure $\mathcal{U}$ but they will differ in the way they order their *Con* constraints. To learn a language is just to learn the ranking of constraints. The rest is inborn.

[18] Note that the ranking is always on the *Con* set. The *Gen* and the *Structural* constraints, are inviolable. They are always fulfilled and not rankeable. In every candidate comparison, the *Gen* and the *Structural* constraints are assumed fulfilled.

| input: $/V^1C^2V^3C^4/ \rightarrow$ | $Onset \gg$ | $NoCoda \gg$ | $Fill^V \gg$ | $Parse \gg$ | $Fill^C$ |
|---|---|---|---|---|---|
| $c_1$: $.\square V^1.C^2V^3.\langle C\rangle$ | | | | * | * |
| $c_2$: $\langle V\rangle.C^2V^3.\langle C\rangle$ | | | | ** | |
| $c_3$: $\langle V\rangle.C^2V^3.C^4\square.$ | | | * | * | |
| $c_4$: $.V^1.C^2V^3C^4.$ | * | * | | | |

Table 2.2: Candidate comparison in H'

## 2.10 Appendix: The set of universal constraints *Con*

OT recognizes two fundamental classes of *Con* constraints: markedness constraints and faithfulness constraints. Faithfulness constraints evaluate the preservation of underlying forms in the output candidates, that is, in the output of the *Gen* function. Markedness constraints assess only to such outputs. These two types of universal constraints are always present in the set *Con*.

OT requires that every OT grammar has markedness and faithfulness constraints. That does not mean that each OT grammar will have the *same* specific constraints of those types. Different OT theories can propose distinct faithfulness constraints and distinct markedness constraints. For example, in a theory for avoiding hiatus,[19] the following *Con* constraints may be assumed.

**For avoiding Hiatus**
(i) *Markedness* constraints:
*HIATUS: Hiatus are prohibited.
(ii)*Faithfulness* constraints:
MAX(V): Vowels may not be deleted.
DEP(C): Consonants epenthesis is prohibited.

While in a basic syllable theory as that exposed in [20], the following constraints could be postulated:

**For a syllable theory in OT**
(i)*Markedness* constraints:
ONSET: A syllable must have an onset.
−CODA: A syllable must not have a coda.
(ii)*Faithfulness*
PARSE: Underlying segments must be parsed into syllable structure (Avoid deletion).

[19] "Hiatus is the phonetic result of the immediate adjacency of vocalic syllable peaks" ([14]:5). The constraints mentioned with relation to hiatus were also taken from [14].

FILL: Underlying segments must be filled with underlying segments (Avoid epenthesis). [20]

The relation between constraints is in principle conflictive. This aspect is related to its violability. Markedness and faithfulness are two antagonistic classes. While faithfulness asks for preservation in the expression of the material given in the underlying form, markedness usually modifies the expression or output candidate, by deleting or aggregating material.

In OT violability does not mean ungrammaticality. Given the conflictive character of markedness and faithfulness, an output is in most situations forced to violate one class. This conflict will be resolved in the interaction of constraints in a particular ranking. In this interaction Eval will select the candidate with the least serious violations, that is, the *optimal* candidate. The *optimal* candidate is not the candidate with null violations, but that candidate that compared with other candidates is the best in satisfying the set of constraints. Grammaticality in OT means *optimality* not inviolability.

[20] In the syllable theory we have presented in 2.9, *Onset* and *NoCoda* are the *markedness* constraints. *Parse*, $Fill^V$ and $Fill^V$ are the *faithfulness* constraints.

# Chapter 3

# CV$_{net}$: A network embodying OT

## 3.1 The 'abstract genome' project

It seems that we can distinguish between two categories. On the one side, the group of devices related with material of genome. In the other, the group of devices concerning nets. In [25] is assumed that each level of the genomic side encodes a level of the net's side. In each side three levels are distinguished.

The genomic side is constituted by the levels: biological genome, abstract genome, innate constraints. Each of these ones will encode respectively the following levels in the net's side: biological neural net, connectionist net, symbolic system. We will call this way of relation among the levels the 'horizontal' relation (see figure 3.1).

In [25] the necessity of introducing an intermediate level between the biological genome's level and the innate constraints' level is suggested . This intermediate level is called 'abstract genome'.

The levels mentioned will indicate a 'vertical' relation. In the genomic side, is assumed that the biological genome's level instantiates the abstract genome's level, and this one, the innate constraints level. In the nets' level, the biological neural net's level instantiates the connectionist net level and this one, the symbolic system level.

In terms of what we treat in this thesis the scheme explained above means the following: $\mathcal{U}$ is in the innate constraint level. To OT corresponds the symbolic system level. That means, $\mathcal{U}$ encodes OT. However, also means in the 'vertical' relation, that $\mathcal{U}$ is instantied in an abstract genome, and this one, in a biological genome. In the nets' side, it means that OT is instantiated in $CV_{net}$, and this network in a biological network.

$CV_{net}$ is in the connectionist net's level. According to [25] it is in this level where a device that encodes such connectionist network was needed.

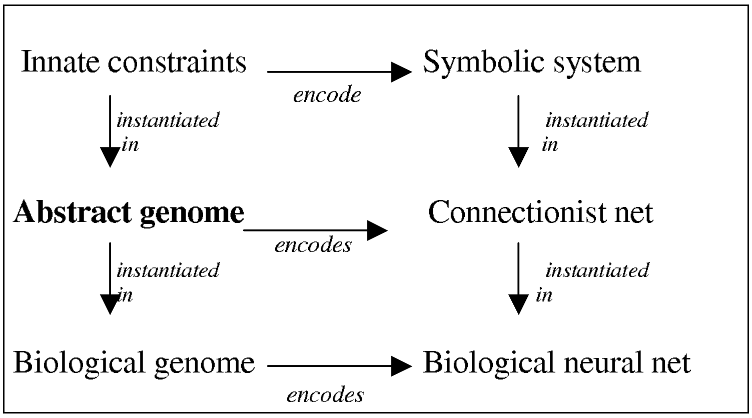


Figure 3.1: Abstract Genome

Their suggestion has been an abstract genome. We agree with the necessity of an intermediate level, but instead of the abstract genome, we introduce penalty logic. For us, penalty logic in the 'horizontal' relation will encode $CV_{net}$ . In the 'vertical' path it instanties the innate constraints' level, in particular, the structure $\mathcal{U}$; and it is instantiated in a biological genome.

## 3.2 In which sense are the *net*-constraints universal?

In OT, constraints are universal as they form part of the structure $\mathcal{U}$. In the connectionist level, constraints are usually characterized as universal when their weights are bound. In this thesis, we assume such a notion of universality in the connectionist level.

In order to avoid confusion, we will use the following names for the 'constraints': Constraints in the level of OT will be called 'OT-constraints', while constraints in the connectionist level will be denominated '*net*-constraints'. This distinction will be extended to the particular constraints. For example, we use the name 'OT-*Onset*' for talking about *Onset* in the linguistic level, and *net-Onset* for *Onset* in the connectionist level.

As we have said, all the *net*-constraints are universal in the sense that all of them assume bound or identical weights. But analogous to the distinction between violable and inviolable constraints given in OT, the *net*-constraints will be divided into constraints with *variable* weights and constraints with *fixed* weights. The *net-Con* constraints will belong to the first category, non-fixed weights. The *net-Gen* and *net-Structural* constraints to the second one, with fixed weights.

## 3.3 CV$_{net}$ architecture

In the CV Syllable Theory, the grammar defines a mapping from an underlying string composed with C's and V's, to a surface form with C's and V's. This mapping is implemented in $CV_{net}$.

$CV_{net}$ is a symmetric local connectionist network that can represent interpretations and expressions of C's and V's.

In $CV_{net}$ we find three layers: an input, an output and a correspondence layer. [1] Take figure 3 as a matrix[2]. The input unit is represented by cells $a_{12}$,$a_{13}$, $a_{14}$, the output layer by cells $a_{21}$, $a_{31}$, $a_{41}$ and $a_{51}$, while the correspondence layer is spread over the remaining cells, in the center of the $CV_{net}$ picture.

In the input and correspondence layers, units are classified by being vowels or consonants. Vowels are denoted by $\nabla$ and consonants by $\bigcirc$. In the output layer a more detailed distinction is introduced. Consonants are distinguished as onsets or codas. In the output layer we have three units: vowels, onset consonants and coda consonants. These are denoted respectively by $\nabla$, $\bigcirc$ and a crescent.

It is important to note that as $CV_{net}$ is a symmetric network, its connectivity pattern has a symmetic feedback. That is, for two units $\alpha$, $\beta$ the weight from $\alpha$ to $\beta$ is equal to the weight from $\beta$ to $\alpha$.

Let us explain in more detail the activity in $CV_{net}$. Suppose that we have the input string /$V^1C^2C^3$/ (grey part of figure 3.2). This input is depicted in the network through activation or disactivation of units in the input layer. Black nodes represent activated nodes, white nodes disactivated nodes. The input /$V^1C^2C^3$/ is depicted in figure 3.2 in the input layer, that is, cells $a_{12}$,$a_{13}$, $a_{14}$. As we have pointed out, in the input layer we only find vowels and consonants. If we think in the OT framework, this may be intepreted as entering as inputs just CV syllables, that is, a syllable that has an onset consonant, a nucleus, but no coda. Graphically, this interpretation seems to be supported too, if you look at figure 3.2, a $\bigcirc$ is found in the input and output layers, and in the last one is interpreted as onset consonant.

In the correspondence layer we still find the distinction of units into vowels and consonants. But in this layer, consonants are not distinguished yet by onset or coda consonants. Let us see how the activity in the correspondence layer goes. The $V^1$ in the input (cell $a_{12}$) is in correspondence with the V in the cell $a_{32}$. Notice that these two V's are activated. Now, if you look at cell $a_{31}$, there is also a V active. This means that the $V^1$ in the input has been parsed, additionally it indicates in which output position the $V^1$ has been parsed. In this case, it has been parsed in the second position of the output. The position of a vowel or consonant in the output will be

[1] In the standard representation of networks, the input, correspondence and output layers will be the rows of input, hidden and output units respectively.

[2] In a cell $a_{ij}$, $i$ represents the row and $j$ the column.

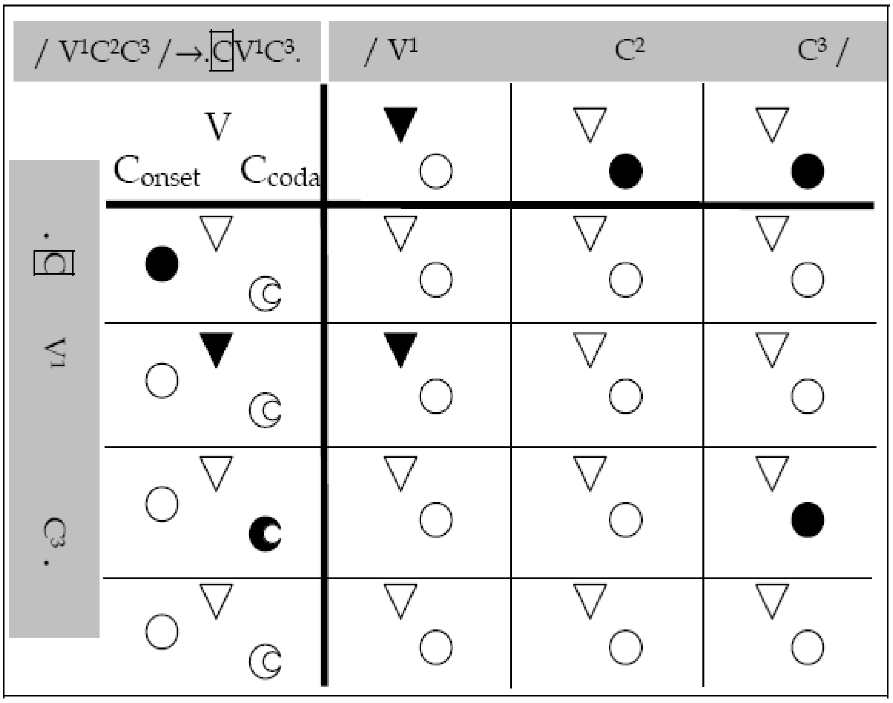


Figure 3.2: $CV_{net}$

relevant for the intepretation if the syllable has onset, nucleus or coda.

Now consider the C$^2$ in cell $a_{13}$. This consonant is active but there is no unit beneath it that indicates to us a correspondence relation. The C in the output (cell $a_{21}$) cannot be interpreted as the consonant C$^2$ in the input (cell $a_{13}$) because in the cell $a_{23}$ the consonant unit is not active. Furthermore, observe that no consonant in the cells to the right of the cell $a_{21}$ is active (cells $a_{22}$, $a_{23}$, $a_{24}$). This tell us that the consonant C in the output (cell $a_{21}$) has been added (epenthesis).

Now, consider the last consonant in the input, the consonant C$^3$ (cell $a_{14}$). Beneath it, there is an active consonant (cell $a_{44}$) and in the output layer, cell $a_{41}$, there is a C-coda active. This means that the C$^3$ has been parsed to a consonant coda in the output.

Notice that in the explanatin given above, we were always looking to relate the vowels in some layer only with vowels in other parts of the net. Similarly, we relate consonants in some layer only with consonants in other layers. This way of proceeding is justified by the *net-Identity* constraint.[3]

## 3.4 The constraints of $CV_{net}$

### 3.4.1 The *net-Con* constraints

In order to describe properly the behavior of the *net-Con* constraints, we have to state that all optimal patterns of activations satisfy the *net-Gen* and the *net-Structural* constraints. The statement is plausible under the assumption that these last sets of *net*-constraints have higher strength than the set of *net-Con* constraints. Usually an exponentially higher weight is assigned to the *net-Gen* and *net-Structural* constraints. This way of proceeding even though it is not adequate because such constraints are not rankeable, in practice guarantees the submission of the violable *net*-constraints.

Below we present the pictures of the *net*-constraints. In these pictures you will see some numbers assigned to each *net*-constraint. These numbers represent the bias values. Strengths are not determined. Connections coefficients are only introduced in the description of each *net*-constraint, but not in the figures.

**a) *net-Onset* and *net-NoCoda***

*net-Onset* is realized through a negative bias $-1$ on every V-unit in the output. In order to avoid this negative value, a positive connection coefficient $+1$ between every V-unit and C-onset unit in the output is required (see figure 3.3).

For *net-NoCoda* a bias $-1$ is related to each C-coda unit. Given that in a CV Syllable Theory codas should be avoided, for this negative bias

[3] This $n$-constraint will be introduced in a later section.

we do not require a positive connection coefficient +1, instead a connection coefficient −1 is introduced (see figure 3.4).

**b) *net-Parse***
In OT faithfulness constraints, there is a correspondence between the elements in the input and those in the output. In order to implement such a correspondence in the connectionist level, correspondence units are needed.

In *net-Parse*, the values of correspondence units plus the values of the bias coefficients lead to a 'zero net Harmony'. In other words, for any *Parse* net, a bias coefficient of −1 associated with each input unit, is assumed. A bias coefficient of −3 is assigned to each correspondence node and +2 for all connection coefficients. From the bias deficit plus a correspondence unit linked to it, we have $(-1) + (-\ 3) = -4$. By the addition of the connection coefficients we have $2 + 2 = 4$. Then $4 + (-4) = 0$. In figure 3.5 only the bias coefficients are represented. Value of connection coefficients are not included.

**c) *net-Fill$^V$* and *net-Fill$^C$***
Similarly to *net-Parse*, for these *net*-constraints connection coefficients will take the value +2, and the bias coefficient in each correspondence unit will be equal to −3. Differently from *net-Parse*, *net-Fill$^V$* and *net-Fill$^C$* do not have bias coefficients in input units but in output units. The bias coefficient in each output unit will be −1. Similarly to *net-Parse*, these constraints lead to a 'zero net Harmony'.

In figure 3.6 both constraints are depicted: *net-Fill$^V$* is constituted by all V-units (see dotted lines) while *net-Fill$^C$* is constituted by all the consonants being onset or coda (see solid lines).

### 3.4.2 The *net-Gen* and *net-Structural* constraints

The *net-Gen* and *net-Structural* constraints are the *inviolable* constraints of $CV_{net}$. More properly stated, these *net*-constraints have fixed strengths. Besides, such *net*-constraints do not change their strengths with respect to each other, so, we can consider them as a one complex constraint whose weight is exponentially larger than any *net-Con* constraint.

We have chosen only some of these constraints for exposition. This decision is due to their inviolable character. Even when these constraints are part of the genomic encoding, they do not permit us to exemplify how from a genomic encoding an *optimal* candidate arises. For that reason, even when they are an important part of $\mathcal{U}$ we will not treat them in detail.

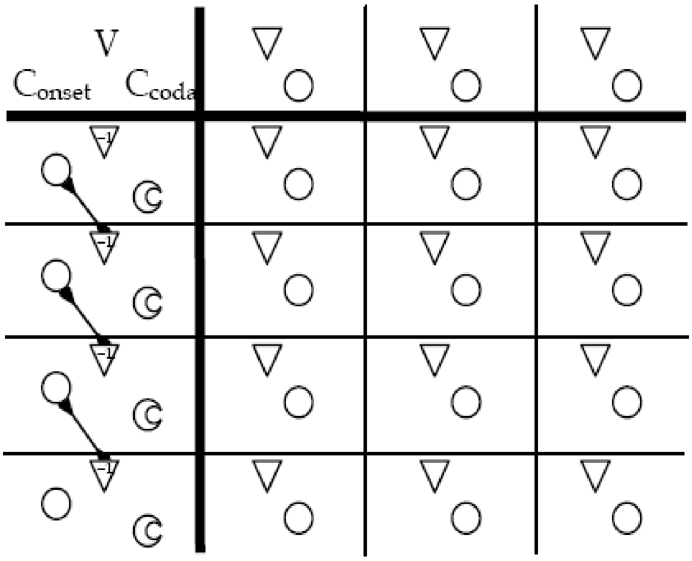


Figure 3.3: *net-Onset*

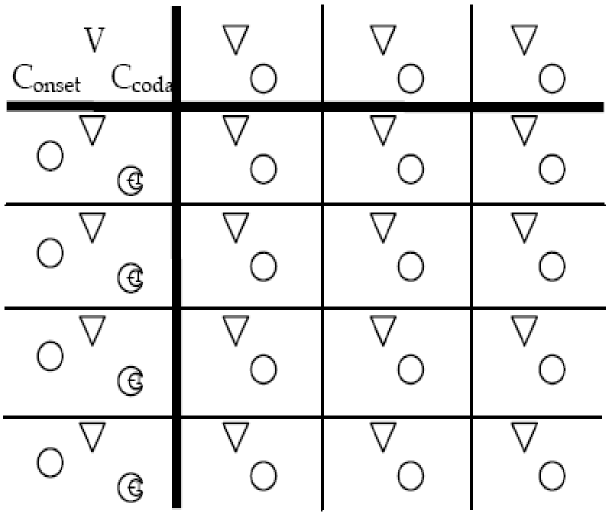


Figure 3.4: *net-NoCoda*

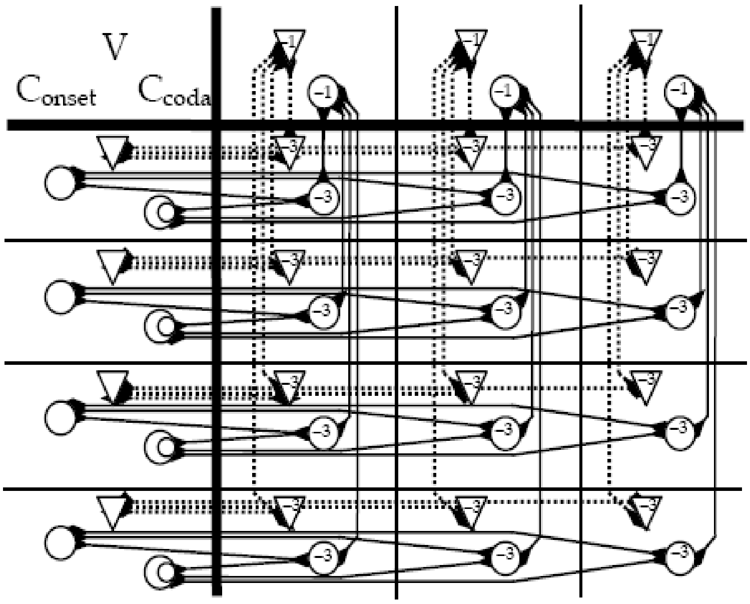


Figure 3.5: *net-Parse*

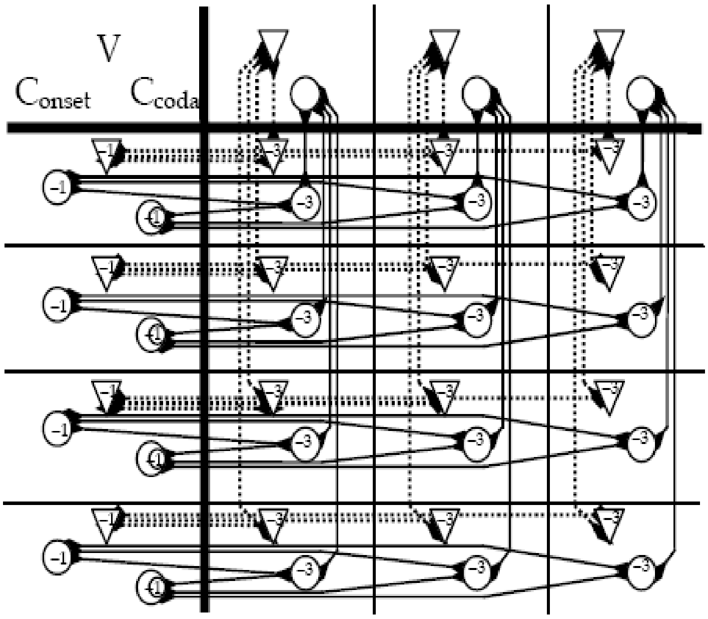


Figure 3.6: *net-Fill*$^V$ and *net-Fill*$^C$

### *net-Gen constraints*

**a) *net-Identity***
For *net-Identity* there is a connection coefficient $-1$ between the C's and V's of the correspondence units. This connection coefficient will avoid the activation of a C and V at the same time supporting in this way the mapping C-C and V-V from input to output (see figure 3.7).

**b) *net-Linearity***
In OT, *linearity* was the constraint that regulated that the order of the elements in the input was preserved in the output. In the connectionist level, this constraint is implemented through the assignment of a value -1 to each connection coefficient between the correspondence units. Observe that in figure 3.8 below, the correspondence units are linked in a way that preserves linear order. Consonants and vowels in a certain cell $a_{ij}$ will be linked with a cell that is in a previous column $k$ and a posterior row $l$.

### *net-Structural constraints*

**c) *net-OutputIdentity***
For this constraint there is a connection coefficient $-1$ between the triple formed in each output cell by the vowels, onset and coda consonants. This negative connection is given in order to avoid the effect that more than one unit of the triple is activated. The ranking of this constraint is required to be significantly higher than the other *net-Gen* or *net-Structural* constraints. This requirement is there because other fixed *net*-constraints will force more than one unit in the output's triple to be actived ([25]:992-3). See figure 3.9.

**d) *net-Correspondence***
The wiring of this *net*-constraint is the same as that of *net-Parse* or of *net-Fills.* The difference arises in the fact that there is no bias either for input or for output units. Correspondence units have bias of $-2$. Connection coefficients get value $+1$ (see figure 3.10).

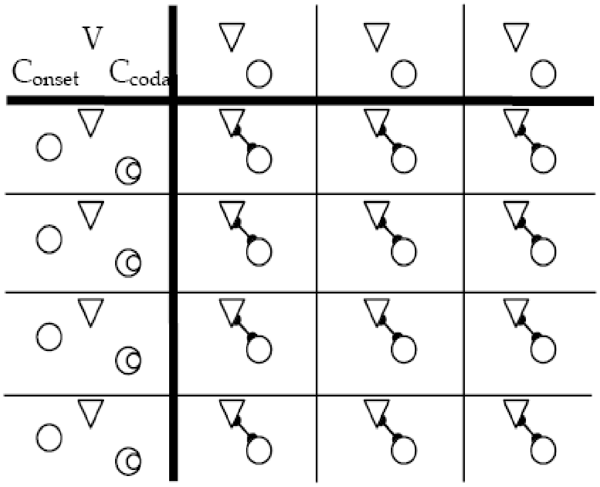


Figure 3.7: *net-Identity*

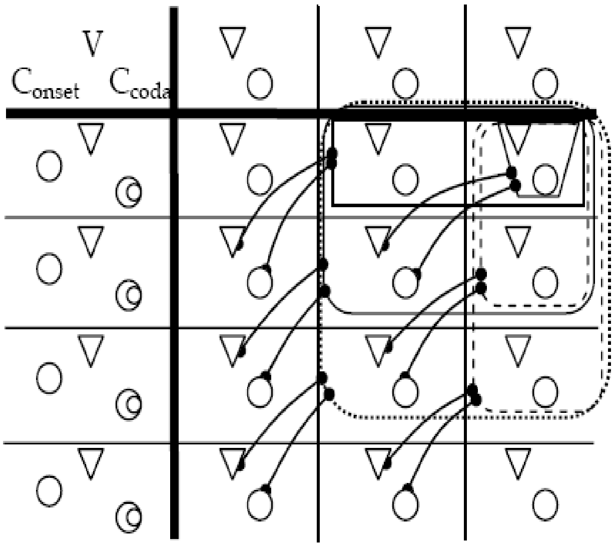


Figure 3.8: *net-Linearity*

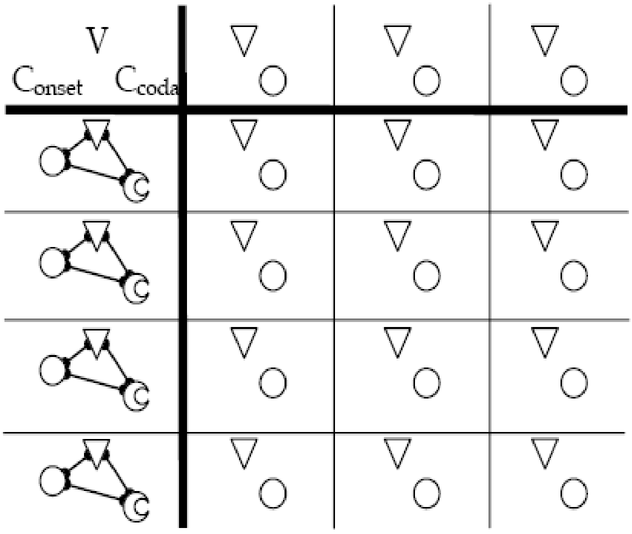


Figure 3.9: *net-OutputIdentity*

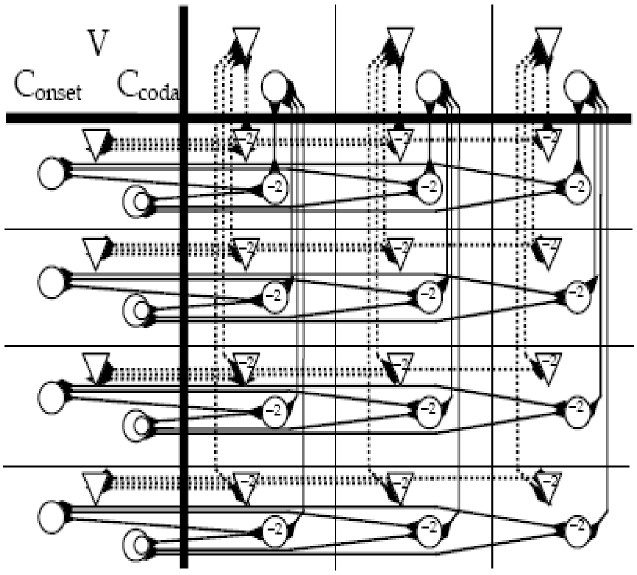


Figure 3.10: *net-Correspondence*

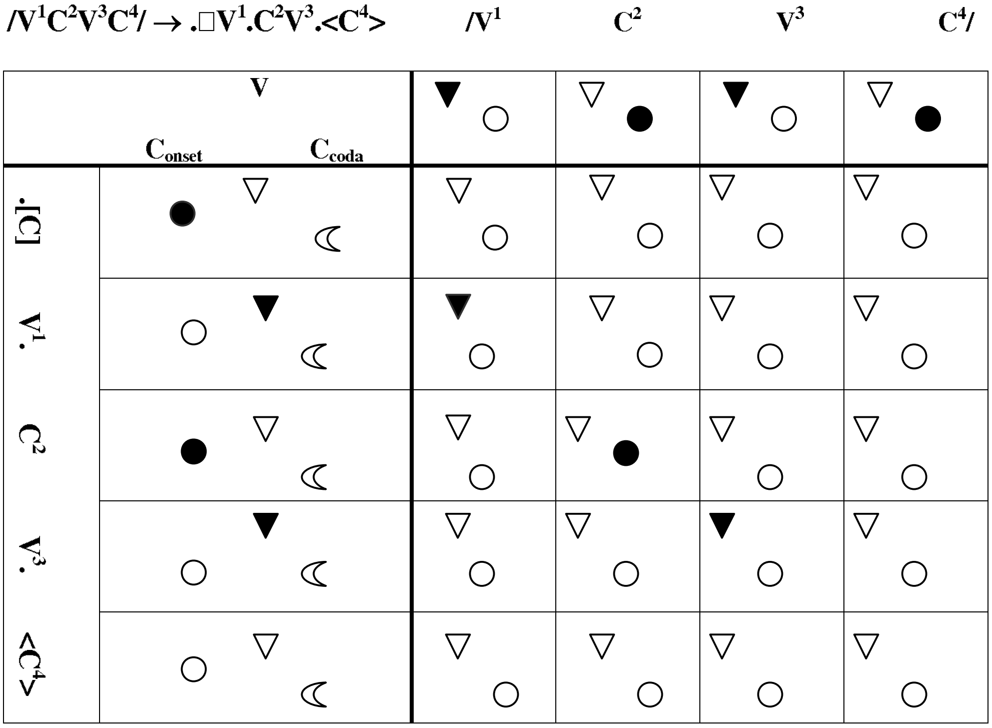


Figure 3.11: Network activity for $c_1$

## 3.5 Candidate Activations

The tables below show the activations of the candidates in table 2.1 (see chapter 2). In order to obtain the *optimal* candidate, besides the activation of the candidates we need exponential weights for introducing the ranking of constraints *Con*.

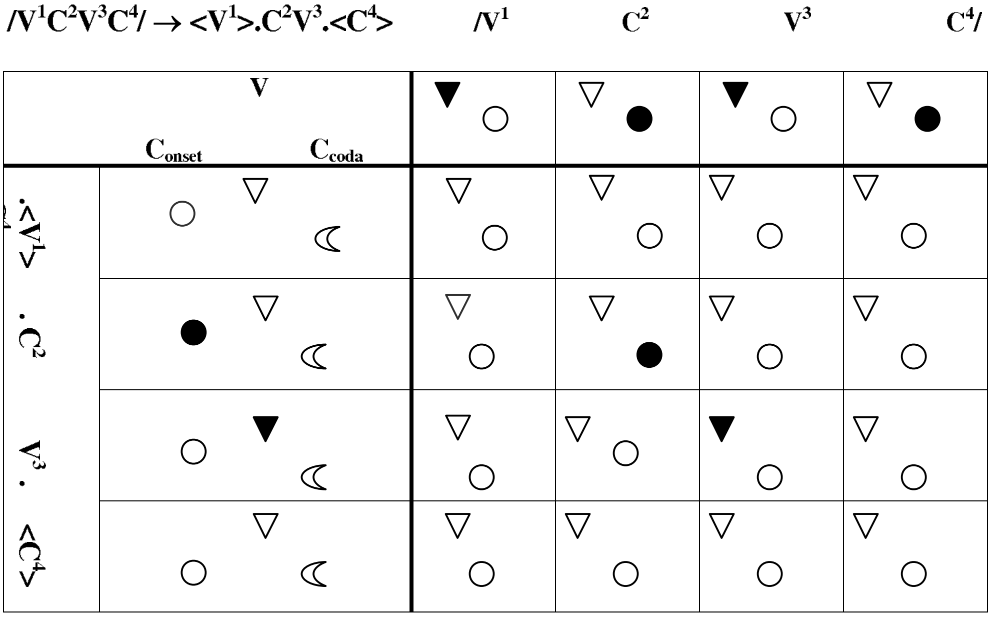


Figure 3.12: Network activity for $c_2$

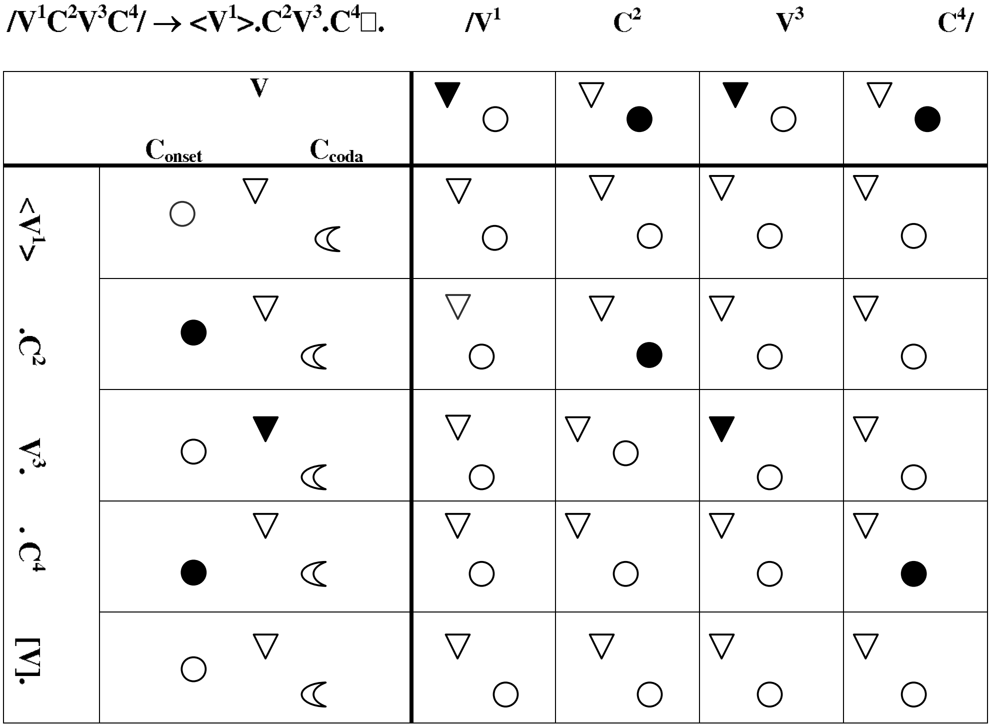


Figure 3.13: Network activity for $c_3$

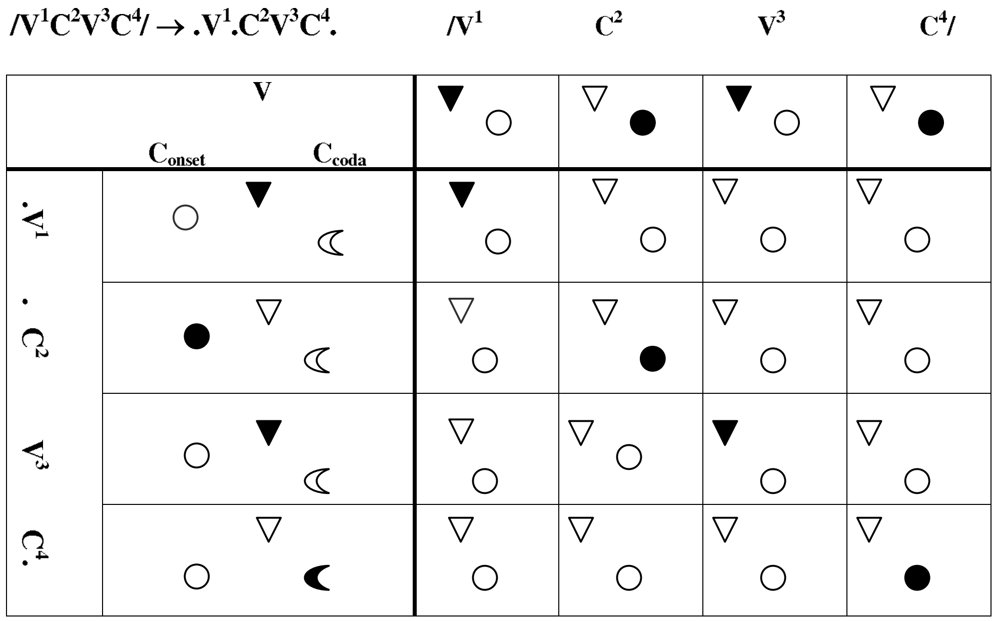


Figure 3.14: Network activity for c$_4$

## 3.6 The energy function E of *net*-constraints

Consider the function:

$$(F1) \qquad E(s) = -\sum_{i<j} w_{ij} s_i s_j$$

This function, called Ljapunov or *energy* function will help us to calculate the value of E relatively to different activation states. Here, we will use for calculating the *minimal energy* of each of the *net*-constraints. Below we present pictures of *net*-constraints links. These links we will call them *micro-constraints*. Below we also introduce the corresponding tables of the energy E of the *micro-constraints*.

*Microconstraints* are important because they simplify the calculation of the energy function E of the *net*-constraints. The character of localist network of $CV_{net}$ permits us to focus in the minimal significative part of the network (the *microconstraints*) and to spread the results to more complex networks build with these parts. If you observe the figures of *net*-constraints, you will see the repetition of the same relations (*microconstraints*) through all the network.

Instead of using the names of s$_1$, s$_2$,... s$_n$ for naming the states in the networks, we have preferred a more easy association with chapter 5, by using similar names to those that we will use in that chapter. We use the names

|  | $s$ |  |  | E($s$) |  | $s$ |  |  | E($s$) |  | $s$ |  |  | E($s$) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | $co_j$ | $v_{j'}$ | $b_0$ |  |  | $co_j$ | $v_{j'}$ | $b_0$ |  |  | $co_j$ | $v_{j'}$ | $b_0$ |  |
| [1] | 1 | 1 | 1 | 0 | [10] | $-1$ | 1 | 1 | 2 | [19] | 0 | 1 | 1 | 1 |
| [2] | 1 | 1 | $-1$ | **$-2$** | [11] | $-1$ | 1 | $-1$ | 0 | [20] | 0 | 1 | $-1$ | $-1$ |
| [3] | 1 | 1 | 0 | $-1$ | [12] | $-1$ | 1 | 0 | 0 | [21] | 0 | 1 | 0 | 0 |
| [4] | 1 | $-1$ | 1 | 0 | [13] | $-1$ | $-1$ | 1 | **$-2$** | [22] | 0 | $-1$ | 1 | $-1$ |
| [5] | 1 | $-1$ | $-1$ | 2 | [14] | $-1$ | $-1$ | $-1$ | 0 | [23] | 0 | $-1$ | $-1$ | 1 |
| [6] | 1 | $-1$ | 0 | 1 | [15] | $-1$ | $-1$ | 0 | $-1$ | [24] | 0 | $-1$ | 0 | 0 |
| [7] | 1 | 0 | 1 | 0 | [16] | $-1$ | 0 | 1 | 0 | [25] | 0 | 0 | 1 | 0 |
| [8] | 1 | 0 | $-1$ | 0 | [17] | $-1$ | 0 | $-1$ | 0 | [26] | 0 | 0 | $-1$ | 0 |
| [9] | 1 | 0 | 0 | 0 | [18] | $-1$ | 0 | 0 | 0 | [27] | 0 | 0 | 0 | 0 |

Table 3.1: Energy *micro-Onset*

*co* for a unit representing an onset consonant, *cc* for a coda consonant, *v* for a vowel and *b* for a bias. For an easier visual relation, we use circles for onset consonants, crescents for coda consonants, triangles for vowels and circles with a an internal cross for biases.

As we mentioned in 3.4.1, for the *net-Onset* constraint the connection coefficient is always positive between the corresponding onset consonant and the vowel. The bias for this constraint is always on the vowel and is negative. In the *micro-Onset* we take the connection coefficient value as the value of the link between the onset consonant and the vowel. The bias value is $-1$. We can depict the network of the *micro-constraint Onset* as follows:

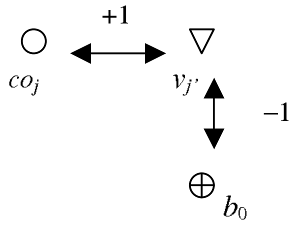


Figure 3.15: *micro-Onset*

The energy function for this system is: $\mathrm{E}(s) = -\ co_j v_{j'} + v_{j'} b_0$. States in rows [2] and [13] show the *minimal energy* E with $-\ 2$ (see table 3.1). [4]

For the *net-NoCoda* constraint each coda unit is affected by a negative bias. In the *micro-Onset* we take the value of such a bias ($-1$) for the

[4] Observe figure 3.3 of *net-Onset*. The onset consonant (circle) is in a previous row with respect to the vowel (triangle). For that reason we use $j$ and $j$' assuming that $j \geq j$'.

representation of the link between the coda consonant and bias unit. The *micro-constraint NoCoda* may be depicted as:

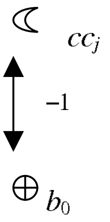


Figure 3.16: *micro-NoCoda*

The energy function E for *NoCoda* would be: $\mathrm{E}(s) = -\ cc_j b_0$. States in [2] and [4] show the *minimal energy* E.

|  | $s$ |  | E($s$) |
|---|---|---|---|
|  | $cc_j$ | $b_0$ |  |
| [1] | 1 | 1 | 1 |
| [2] | 1 | −1 | − 1 |
| [3] | 1 | 0 | 0 |
| [4] | −1 | 1 | −1 |
| [5] | −1 | −1 | 1 |
| [6] | −1 | 0 | 0 |
| [7] | 0 | 1 | 0 |
| [8] | 0 | −1 | 0 |
| [9] | 0 | 0 | 0 |

Table 3.2: Energy *micro-NoCoda*

For the *net-Parse* constraint the connection coefficients are always positive. There are two kind of biases, biases on the input units (−1), and biases on the correspondence units (+3). In figure 3.17 and 3.18, we distinguish input units with an upper-index, while output units has a single-down index. Correspondence units have a double-down index. In figure 3.17, we depict the micro-constraint *Parse*.

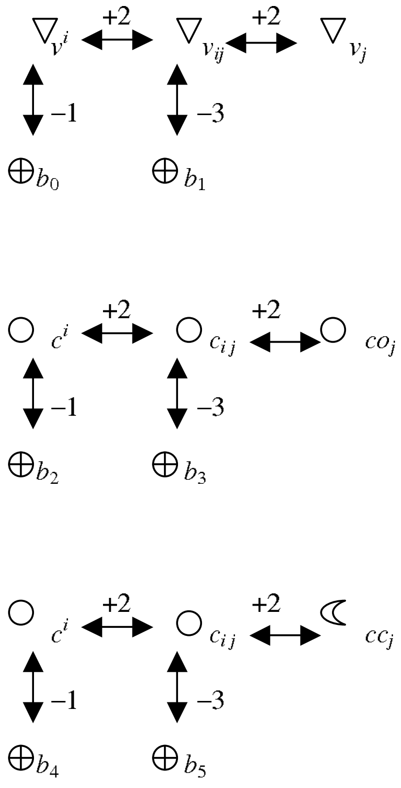


Figure 3.17: *micro-Parse*

The energy function for this system is: E$(s) = -\ 2v^i v_{ij}$-$2v_{ij}v_j + v^i b_0 + 3v_{ij}b_1$. States in rows [4] and [29] show the *minimal energy* E with $-\ 8$ (see table 3.3). For brevity, here we only show the cases corresponding to activated and inactivated states. These will be the relevant ones for the translation into penalty logic.

The *micro*-constraints $Fill^V$ may be depicted as follows:

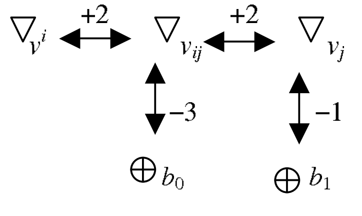
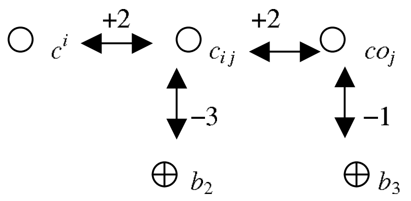


Figure 3.18: micro-$Fill^V$

|  | $s$ |  |  |  |  | E($s$) |
|---|---|---|---|---|---|---|
|  | $v^i$ | $v_{ij}$ | $v_j$ | $b_0$ | $b_1$ |  |
| [1] | 1 | 1 | 1 | 1 | 1 | 0 |
| [2] | 1 | 1 | 1 | 1 | −1 | −6 |
| [3] | 1 | 1 | 1 | −1 | 1 | −2 |
| [4] | 1 | 1 | 1 | −1 | −1 | $-\mathbf{8}$ |
| [5] | 1 | 1 | −1 | 1 | 1 | 0 |
| [6] | 1 | 1 | −1 | 1 | −1 | −6 |
| [7] | 1 | 1 | −1 | −1 | 1 | −2 |
| [8] | 1 | 1 | −1 | −1 | −1 | −4 |
| [9] | 1 | −1 | 1 | 1 | 1 | 2 |
| [10] | 1 | −1 | 1 | 1 | −1 | 8 |
| [11] | 1 | −1 | 1 | −1 | 1 | 0 |
| [12] | 1 | −1 | 1 | −1 | −1 | 6 |
| [13] | 1 | −1 | −1 | 1 | 1 | −2 |
| [14] | 1 | −1 | −1 | 1 | −1 | 0 |
| [15] | 1 | −1 | −1 | −1 | 1 | −4 |
| [16] | 1 | −1 | −1 | −1 | −1 | 2 |
| [17] | −1 | 1 | 1 | 1 | 1 | 2 |
| [18] | −1 | 1 | 1 | 1 | −1 | −4 |
| [19] | −1 | 1 | 1 | −1 | 1 | 4 |
| [20] | −1 | 1 | 1 | −1 | −1 | −2 |
| [21] | −1 | 1 | −1 | 1 | 1 | 6 |
| [22] | −1 | 1 | −1 | 1 | −1 | 0 |
| [23] | −1 | 1 | −1 | −1 | 1 | 8 |
| [24] | −1 | 1 | −1 | −1 | −1 | 2 |
| [25] | −1 | −1 | 1 | 1 | 1 | −4 |
| [26] | −1 | −1 | 1 | 1 | −1 | 2 |
| [27] | −1 | −1 | 1 | −1 | 1 | −2 |
| [28] | −1 | −1 | 1 | −1 | −1 | 4 |
| [29] | −1 | −1 | −1 | 1 | 1 | $-\mathbf{8}$ |
| [30] | −1 | −1 | −1 | 1 | −1 | −2 |
| [31] | −1 | −1 | −1 | −1 | 1 | −6 |
| [32] | −1 | −1 | −1 | −1 | −1 | 0 |

Table 3.3: *Energy micro-Parse*

$Fill^V$ may be depicted as:



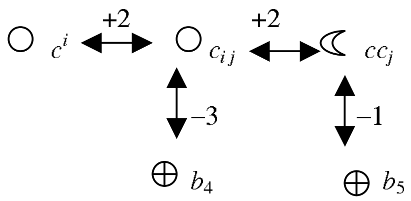


Figure 3.19: micro-$Fill^C$

The energy calculation for these constraints is similar to those of *micro-Parse*.

## 3.7 Summary

In this chapter we have introduced the network $CV_{net}$. For doing that we have presented the *net*-constraints, the *micro*-constraints, the energy function E and activations for particular candidates. All these topics are related.

For explaining such a relation, first, we will display $CV_{net}$ in a more conventional way of representing networks, that is, a network in which we have rows of input units, of hidden units and of output units. In such a network, we introduce arrows whose direction is forward and backward due to the symmetry of $CV_{net}$.

Look at figure 3.20. In this representation, the first row from bottom to top represents the input units of $CV_{net}$. The row in the middle contains the hidden or correspondence units. The row at the top is constituted by the output units.

In the input row, we have two kind of units, vowels represented by triangles and consonants represented by circles. Observe that in this row, each circle is related to a circle in the hidden row and each triangle is related to a triangle in the hidden row.

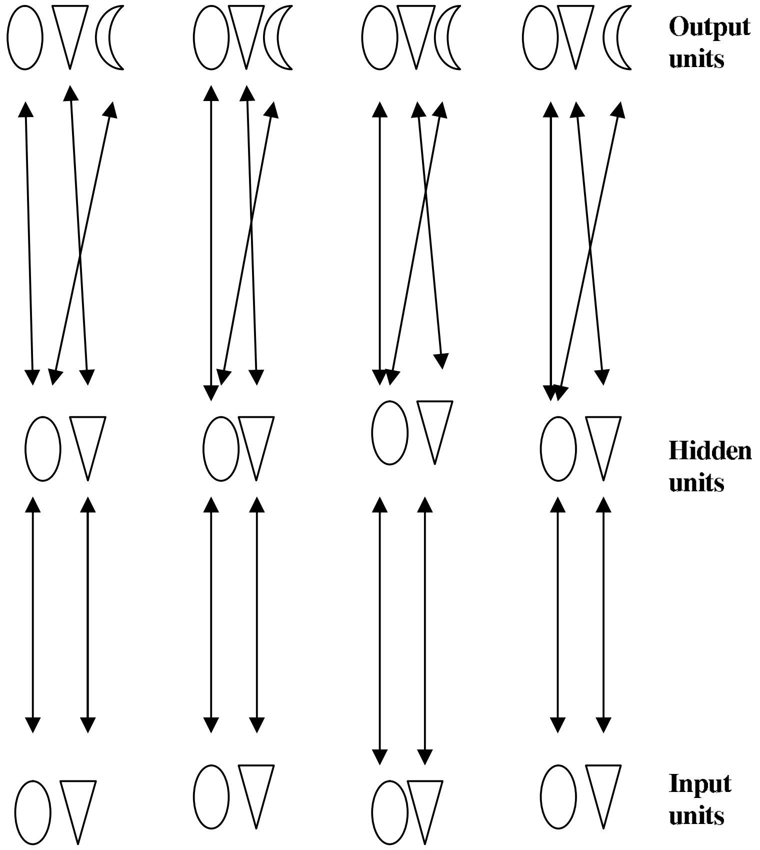


Figure 3.20: Another representation of $CV_{net}$

In the hidden row we can also see two kind of units, consonants and vowels. From this row, vowels will be related with vowels in the output. Consonants in the hidden row will be related with an onset or a coda consonant in the output row. It is important to note that if the strenght of the activation that comes from the consonant located in the hidden unit is stronger towards the onset consonant than towards the coda consonant, the onset consonant will tend to be active while the coda consonant will tend to be disactive. Similarly, when the activation is stronger towards the coda consonant. The onset consonant will have less activation and it will tend to be disactive.

The *micro-Onset* constraint show their minimal energy E in two cases: (i) where the onset consonant and vowel are active but the bias is not active; and (ii) where the onset consonant and vowel are disactive but the bias is active (see table 3.1). These two cases are the same in which the *net*-Onset constraint is fulfilled. These cases can be represented as in figure 3.21.

For *net-Parse* the states with minimal energy E are those in which the vowels in the input, hidden and output rows are all activated but their biases are not; or when such vowels are disactive but the biases are active (see table 3.3). Similarly with consonants (see figure 3.21).

Observe that in *net-Parse* vowels are active. This would imply according with *net-Onset* that onset consonants must be active. However, according to *net-*$Fill^V$, consonants in the hidden row and in input row become active. Moreover, given that onset consonants in the output row are active, the activation of consonants in the hidden units must be stronger to such units intead of coda units. That is, codas will tend to be disactive. [5] Such activations and disactivations also fulfills the *net-Parse* constraint with respect to consonants and $Fill^C$ constraint. (see figure 3.22 at the top).

Now, consider the network activity of candidate $c_1$. From figure 3.22 (at the bottom) and the representation of the constraint activity we can guess which constraints such a candidate violates. Observe for example, the first unit active in the output (from left to right) is a consonant. Such consonant do not active the corresponding consonants in the hidden and input units. This tells us that candidate c1 is violating the constraint *net-*$Fill^C$. Moreover, if you observe the input row, the consonant located to the most right is active but the corresponding consonants of the hidden and output rows are not active. This can be read as a violation of *net-Parse*.

[5] This fact will fulfill the *net-Nocoda* constraint.

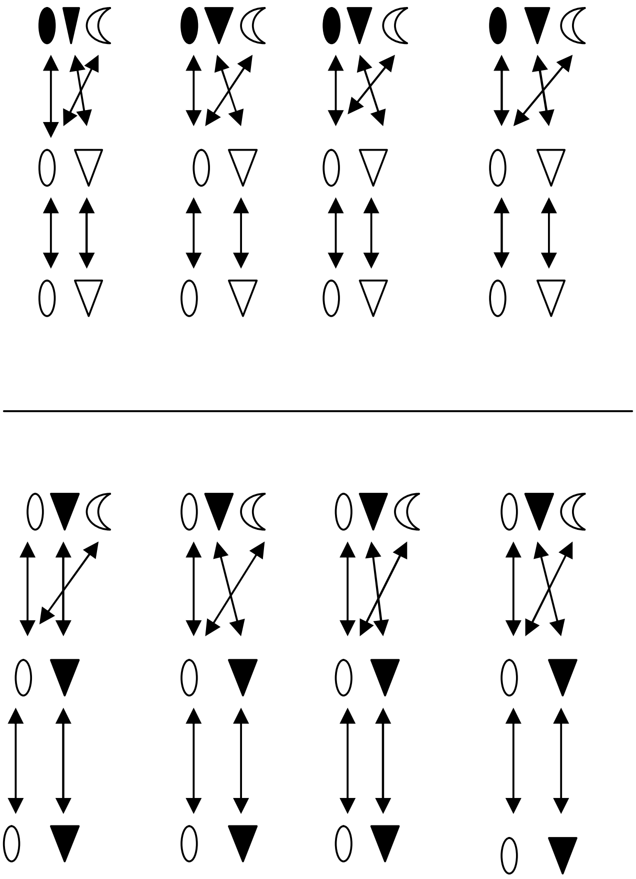


Figure 3.21: *net-Onset and net-Parse*

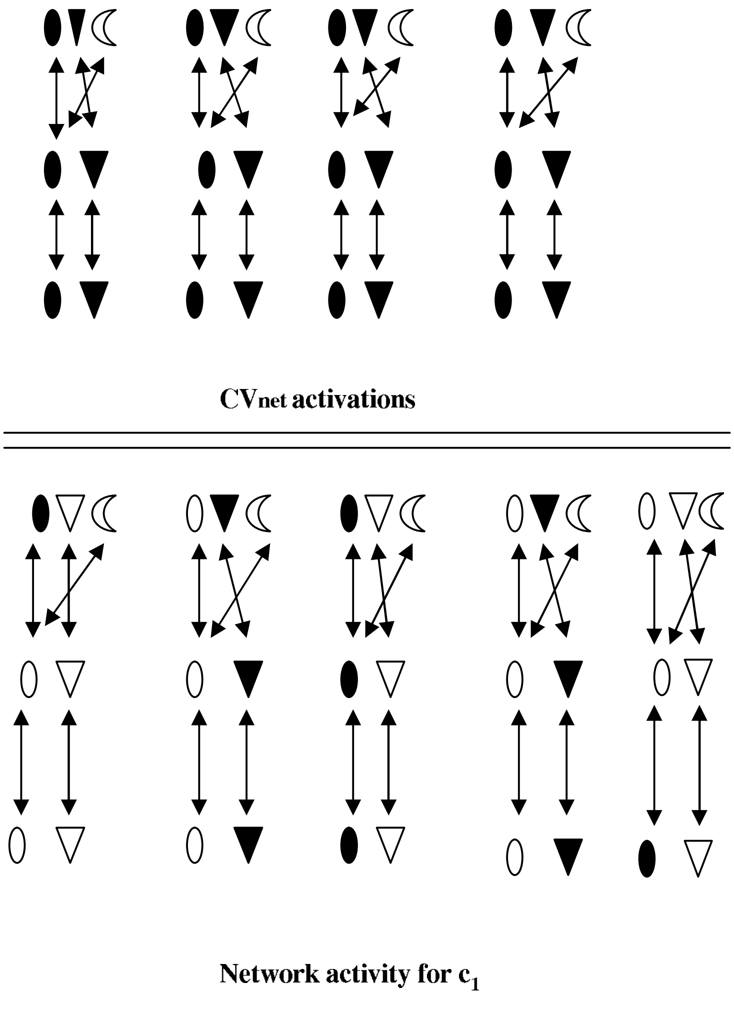


Figure 3.22: $CV_{net}$ and candidate $c_1$ activation

# Chapter 4

# Penalty Logic

## 4.1 Penalty Logic as an abstract genome

According to [25] the abstract genome requires two assumptions in order to encode $CV_{net}$. A general assumption about how the machinery of the network develops and an assumption more specific, about a set of 'abstract' genes that determine a particular network. In the general assumption, it is accepted that the system is capable for example, of distinguishing the various types of units: input, output and correspondence units; vowels, and (onset and coda) consonants; as well as their organization. In the specific assumption, it is accepted that each type of unit of $CV_{net}$ represents a specific set of gens. In this set of gens the necessary information is encoded for drawing the links in the network. In this sense, the output obtained from the artificial network is an expression of our personal genetic information.[1]

In this chapter we introduce penalty logic. As we have mentioned in chapter 3, we placed this logic in the abstract genome's level paying attention that this logic encodes $CV_{net}$. If this is the case, perhaps penalty logic should have similar assumptions to those accepted by an abstract genome.

## 4.2 The Penalty Logic System

Penalty logic, introduced by Pinkas in [19], is an extended propositional calculus that associates to the formulae in a penalty knowledge base, a price or cost to pay if this formula is violated or not satisfied. The price or cost of a formula is its *penalty*, that will be indicated using a real positive number. Penalties induce a ranking among the formulae of a knowledge base, in terms of the most costly or the least costly formula in the base.

Formulae in a penalty knowledge base will be called assumptions or beliefs. The ranking generated by the penalties will indicate to us the strength

[1] We cannot state that the output is totally determined by genetic information, because in such a case we are excluding the possibility of learning the ranking of constraints.

of the beliefs or assumptions in a certain knowledge base. Our weakest beliefs have the least priority, those of which we are least confident, and to which we assign a low price or penalty. In contrast to this, to our stronger beliefs we give a high priority, a high price. If we have to get rid of some formulae in our knowledge base, it is better to exclude the weak beliefs. Retaining weak beliefs will be more costly in comparison with retaining strong beliefs.[2]

An interesting aspect in [19] is that it considers inconsistent knowledge bases. Penalties are used for selecting the preferred consistent subsets in an inconsistent knowledge base. Such selection will induce a nonmonotonic inference relation.

Below we present in a more formal way some of the concepts that we have just mentioned. The definitions in the next section are based on [19] and [4].

DEFINITION 4.1
Let $\mathcal{L}$ be the basic language of the standard propositional calculus. A penalty knowledge base (PKB) $\Psi$ is a set of pairs ($\mathcal{S}$, $\kappa$) where $\mathcal{S}$ is a set of well-formed formulae of $\mathcal{L}$ and $\kappa$ is the penalty function that maps each formula of $\mathcal{S}$ to the set $\{0,\infty\}$, i.e. $\kappa$: $\mathcal{S} \rightarrow \{0,\infty\}$. [3] $\Delta_\Psi$ stands for the set of propositional formulae of $\mathcal{L}$ that are in the PKB $\Psi$.

As we have defined, a PKB contains propositional formulae with their corresponding penalties. The function $\kappa$ guarantees that only one penalty is assigned to each formula of the PKB. Furthermore, $\kappa$ also induces a ranking among the formulae that belong to the PKB. This ranking will permit the selection of the preferred or optimal subsets.

An example of a PKB is the following:

4.1.1 *Example of a PKB*
Suppose that we believe that penguins are birds, that birds fly and penguins do not fly; and the strength of our beliefs is represented respectively by the numbers 20, 5 and 10.

Consider a set P={ $p_0$, $p_1$, ..., $p_n$ } of propositional atoms where $p_i$ stands for ‘animal $i$ is a penguin’. Sets B={ $b_0$, $b_1$, ..., $b_n$ } and F={ $f_0$, $f_1$, ..., $f_n$ } also of propositional atoms, where $b_i$ stands for ‘animal $i$ is a bird’ and $f_i$ for ‘animal $i$ flies’.

[2] To retain a weak belief instead of a strong one means that we have to pay the price for violating the strong belief. This price is higher than the price for a lower belief. In this sense, it is said that retaining a weak belief is more costly than keeping a strong one.

[3] Some authors call a set of sentences plus certain penalties, a ‘penalty knowledge base’. Here, we will assume a broad sense of PKB. A PKB is integrated by formulae, sentences or both. Their penalties can be fixed or variable. In the examples we use in this thesis, we will use PKBs composed by formulae and fixed weights for a better exemplification. Formulae will permit us to modify valuations, and fixed weights to calculate specific costs.

Our PKB will be constituted by[4]:

| | |
|---|---|
| $p_i \rightarrow b_i$ | 20 |
| $b_i \rightarrow f_i$ | 5 |
| $p_i \rightarrow \neg\ f_i$ | 10 |

Call this PKB *ex*. The set of propositional formulae in *ex* is

$$\Delta_{ex} = \{p_i \rightarrow b_i,\, b_i \rightarrow f_i,\, p_i \rightarrow \neg\ f_i \mid i \leq n\ \}$$

***Syntax***

DEFINITION 4.2
Let $\varphi$ be any formula of $\mathcal{L}$. A scenario of $\varphi$ in a PKB $\Psi$ is a consistent subset $\Delta$' of $\Delta_\Psi$ such that $\Delta$' $\cup$ $\{\ \varphi\ \}$ is consistent.[5]
The cost of a scenario $\Delta$' in a PKB $\Psi$, written $\mathcal{K}_\Psi(\Delta$'), is the sum of the penalties of the formulae of $\Psi$ that *do not belong* to $\Delta$'. That is,

$$(F2) \qquad \mathcal{K}_\Psi = \sum_{\delta \in (\Delta_\Psi - \Delta')} \kappa(\delta)$$

where $\kappa(\delta)$ stands for the penalty or cost of an expression $\delta$.[6]

Note that the PKB may be inconsistent where the scenario is not. A scenario of a formula $\varphi$ will be a set of formulae of the PKB that is consistent. This consistency must hold even for the union with $\varphi$.

4.2.1 *Example of scenarios*
Take $\varphi = p_i$ and consider the following subsets of the PKB *ex*:
$\Delta_1 = \{p_i \rightarrow b_i,\, b_i\!\rightarrow f_i\ \}$.
$\Delta_2 = \{p_i \rightarrow b_i,\, p_i\!\rightarrow \neg\ f_i\ \}$.
$\Delta_3 = \{b_i \rightarrow f_i\ \}$.
$\Delta_4 = \{b_i\!\rightarrow f_i,\, p_i\!\rightarrow \neg\ f_i\ \}$.
$\Delta_5 = \Delta_{ex} =\!\{p_i \rightarrow b_i,\, b_i\!\rightarrow f_i,\, p_i\!\rightarrow \neg\ f_i\}$.

The subset $\Delta_1$ is a scenario of $p_i$, because $\Delta_1 \cup \{p_i\}$is a consistent set. The subset $\Delta_2$ is also a scenario of $p_i$, because $\Delta_2 \cup \{p_i\}$ is consistent. $\Delta_3$ and

[4] '$\neg$' and '$\rightarrow$' are as in classical propositional logic.

[5] It is not necessary that $\varphi$ belongs to the PKB $\Psi$. It can be an external element. The requirement asked for $\varphi$ is just its consistency of its union with a subset of the PKB $\Psi$.

[6] With respect to penalty logic, we will use the term 'expression' as referring to a group of signs that represent a concept. Expressions in the penalty logic context should be distinguished from the 'expressions' in an OT context (see definition 2.3).

$\Delta_4$ are also scenarios of $p_i$; the unions $\Delta_3 \cup \{p_i\}$, $\Delta_4 \cup \{p_i\}$ are consistent. But the subset $\Delta_5$ is not a scenario of $p_i$ because $\Delta_5 \cup \{p_i\}$ is inconsistent. Note that we could infer $f_i \wedge \neg fi$.

The formula $p_i$ may be read as 'animal $i$ is a penguin'. To have the belief that a certain animal is a penguin is consistent with the beliefs that penguins are birds and birds fly. This is scenario $\Delta_1$ of $p_i$. We can consistently hold that certain animals are penguins, that penguins are birds but they do not fly. This is scenario $\Delta_2$ of $p_i$. Scenario $\Delta_3$ of $p_i$ is a subset of $\Delta_1$. The belief that a certain animal is a penguin is consistent with the beliefs that birds fly. Besides, we can without inconsistency state that birds fly but penguins do not. Notice that in this case, we do not affirm that penguins are birds. This is scenario $\Delta_4$ of $p_i$. But if we affirm that certain animals are penguins, we cannot hold consistently that penguins are birds, birds fly and penguins do not fly. In this case, penguins would be birds that fly and do not fly. This is $\Delta_5$ of $p_i$.

4.2.2 *Example of cost of scenarios*
The cost of a scenario is the sum of the penalties of the formulae that belong to the PKB but do not belong to the scenario. Consider scenario $\Delta_3$ of $p_i$ introduced above. The formulae of the PKB that do not belong to $\Delta_3$ are $p_i \rightarrow b_i$ and $p_i \rightarrow \neg f_i$, with penalties of 20 and 10 respectively. The sum of these penalties is the cost of $\Delta_3$, in this case, $\mathcal{K}_\Psi(\Delta_3) = 30$.

A more formal way to calculate the cost is following the formula F2. This formula tells us what we have just done: We should add the penalties of the formulae or expressions that belong to the PKB without considering the formulae in the scenario. For calculating the cost of $\Delta_3$ using the formula F2 we need the sum of the penalties of the formulae in $\Delta_{ex}$, i.e., $\sum \kappa(\delta)$ such that $\delta \in \Delta_{ex}$, in this case 35. Besides, we need the sum of the penalties of the formulae in $\Delta_3$, i.e., $\sum \kappa\ (\delta)$ such that $\delta \in \Delta_3$, which is equal to 5. The cost of $\Delta_3$ with respect to the PKB *ex* appears by the substraction of the second sum from the first one. That is, $\mathcal{K}_{ex}(\Delta_1) = 35 - 5 = 30$.

DEFINITION 4.3
The scenarios of a formula $\varphi$ that have a minimal cost $\mathcal{K}$ with respect (w.r.t.) to a PKB $\Psi$ are the *optimal* of $\varphi$ w.r.t. $\Psi$.

Consider the scenarios of $p_i$ w.r.t. the PKB *ex* introduced in 4.2.1. The cost of these scenarios is:

$\mathcal{K}_{ex}\ (\Delta_1) = 10$
$\mathcal{K}_{ex}\ (\Delta_2) = 5$
$\mathcal{K}_{ex}\ (\Delta_3) = 30$
$\mathcal{K}_{ex}\ (\Delta_4) = 20$

According to definition 4.3, the *optimal* scenario of $p_i$ is $\Delta_2$ because it has the minimal cost.

DEFINITION 4.4
Let $\alpha$, $\beta$ be two arbitrary formulae of a PKB $\Psi$. $\alpha \vdash_\Psi \beta$ *if and only if* $\beta$ is an ordinary consequence of every optimal scenario of $\alpha$ in $\Psi$.

$\Delta_2 = \{p_i \rightarrow b_i, p_i \rightarrow \neg\ f_i\ \}$ is the *optimal* scenario of $p_i$ w.r.t. the PKB $\Psi$. From this scenario we can infer for example that $\Delta_2 \cup \{p_i\} \vdash_\Psi b_i$ and $\Delta_2 \cup \{p_i\} \vdash_\Psi \neg\ f_i$. But we cannot infer $\Delta_2 \cup \{p_i\} \vdash_\Psi f_i$.

***Semantics***

DEFINITION 4.5
Let $\nu$ be an interpretation function that maps the well-formed formulae of $\mathcal{L}$ to the set $\{-1,1\}$, i.e. $\nu$: $\mathcal{S} \rightarrow \{-1,1\}$. The *system energy* of the interpretation is calculated by

$$(F3) \qquad \mathcal{E}_\Psi = \sum_{\delta \in \Delta_\Psi, \|\delta\|_\nu = -1} \kappa(\delta)$$

The function $\nu$ maps well-formed propositional formulae to $\{-1,1\}$. The mapping of a formula $\varphi$ to $-1$ can be interpreted as $\varphi$ is violated, wrong, unsatisfied, and so on. In contrast, the mapping of a formula $\varphi$ to 1 can be read as $\varphi$ is not violated, right, satisfied, and so on. The function $\nu$ behaves as the classical valuation in terms of 0 and 1 but in this case $-1$ plays the role of 0.

$F3$ says that in order to calculate the *energy system* of an interpretation $\nu$, we should add the penalties of those expressions or formulae that belong to the PKB $\Psi$ but that are not satisfied under the interpretation $\nu$. For example, if $\nu$ assigns $\nu(b_i \rightarrow f_i) = 1$ and $\nu(p_i \rightarrow b_i) = \nu(p_i \rightarrow \neg\ f_i) = -1$, we sum up the penalties of the formulae $p_i \rightarrow b_i$ and $p_i \rightarrow \neg\ f_i$, that is, 20 + 10. The *energy system* of the interpretation $\nu$ w.r.t. $\Psi$, i.e. $\mathcal{E}_\Psi(\nu)$ would be 30.

If it is the case that all formulae are satisfied under $\nu$, the minimal energy for the interpretation $\nu$ arises. However, if the opposite is the case, all formulae are unsatisfied, a maximal energy of the interpretation $\nu$ arises. In penalty logic, the minimal *system energy* of an interpretation $\nu$ is zero. There is no fixed maximum. The maximal *system energy* of $\nu$ is determined w.r.t. the formulae satisfied in each case. The more satisfied formulae there are, the less energy the intepretation $\nu$ has. The less satisfied formulae (i.e. more unsatisfied), the more energy for $\nu$.

|  |  |  |  | *penalties* |  |  |  |
|---|---|---|---|---|---|---|---|
|  |  |  |  | 20 | 5 | 10 |  |
|  | $p_i$ | $b_i$ | $f_i$ | $p_i \rightarrow b_i$ | $b_i \rightarrow f_i$ | $p_i \rightarrow \neg f_i$ | $\mathcal{E}_{ex}(\nu)$ |
| $\mathcal{M}_1$ | 1 | 1 | 1 | 1 | 1 | $-1$ | 10 |
| $\mathcal{M}_2$ | 1 | 1 | $-1$ | 1 | $-\ 1$ | 1 | 5 |
| $\mathcal{M}_3$ | 1 | $-1$ | 1 | $-1$ | 1 | $-1$ | 30 |
| $\mathcal{M}_4$ | 1 | $-1$ | $-1$ | $-1$ | 1 | 1 | 20 |

Table 4.1: Models of the PKB *ex*

DEFINITION 4.7
Let $\Psi$ be a PKB. The models of $\Psi$ with the minimal energy $\mathcal{E}$ are the *preferred* models of $\Psi$.

In the case of propositional logic a model is just a valuation. Following this, we consider a model to be a certain assignation of values under an intepretation $\nu$. The *preferred models* will be the models with the highest number of satisfied propositions.

Note that the formulae in the PKB *ex* are implications. The truth value of an implication will be calculated in an analogous way to the classical one, but here instead of 0 we use $-1$. That is, $\nu(\varphi \rightarrow \psi) = -1$ *iff* $\nu(\varphi) = 1$ and $\nu(\psi) = -1$, where $\varphi$, $\psi$ are formulae of the standard propositional calculus.

Consider the following models with their respective *system energy*:

In $\mathcal{M}_1$,
$\nu(p_i \rightarrow b_i) = 1$ because $\nu(p_i) = \nu(b_i) = 1$;
$\nu(b_i \rightarrow f_i) = 1$ because $\nu(b_i) = \nu(f_i) = 1$.
But $\nu(p_i \rightarrow \neg\ f_i) = -\ 1$ because $\nu(p_i) = 1$ and $\nu(\neg\ f_i) = -\ 1$.

Notice that in $\mathcal{M}_1$ the formulae $p_i \rightarrow b_i$ and $b_i \rightarrow f_i$ are satisfied. We can think of this model as a world in which birds fly and penguins are birds that can fly. In order to calculate the *system energy* of an intepretation $\nu$, we need to add the formulae not satisfied in this model. The *system energy* of interpretation $\nu$ w.r.t. $\mathcal{M}_1$ is 10. Because the formula not satisifed is $p_i \rightarrow \neg\ f_i$ with penalty 10.

In $\mathcal{M}_2$,
$\nu(p_i \rightarrow b_i) = 1$ because $\nu(p_i) = \nu(b_i) = 1$;
$\nu(b_i \rightarrow f_i) = -1$ because $\nu(b_i) = 1$ and $\nu(f_i) = -1$.
$\nu(p_i \rightarrow \neg\ f_i) = 1$ because $\nu(p_i) = 1$ and $\nu(\neg\ f_i) = 1$.

In $\mathcal{M}_2$ the formulae $p_i \rightarrow b_i$ and $p_i \rightarrow \neg\ f_i$ are satisfied. Model $\mathcal{M}_2$ may be thought of as a world in which penguins are birds that do not fly. Notice that in this world birds do not fly either, $b_i \rightarrow \neg\ f_i = 1$.

The *system energy* of interpretation $\nu$ w.r.t. $\mathcal{M}_2$ is 5. Because the formula that is not satisfied is $\nu(b_i \rightarrow f_i)$ with penalty 5.

In $\mathcal{M}_3$,
$\nu(p_i \rightarrow b_i) = -1$ because $\nu(p_i) = 1$ and $\nu(b_i) = -1$.
$\nu(b_i \rightarrow f_i) = 1$ because $\nu(b_i) = -1$ $\nu(f_i) = 1$; and
$\nu(p_i \rightarrow \neg\, f_i) = -1$ because $\nu(p_i) = 1$ and $\nu(\neg\, f_i) = -1$.

In this model the only formula of the PKB *ex* satisfied is $b_i \rightarrow f_i$, $\mathcal{M}_3$ represents a world in which birds fly. Moreover, a world in which penguins do fly, because $\nu(p_i \rightarrow f_i) = 1$, but are not birds, $\nu(p_i \rightarrow \neg\, b_i) = 1$.

The *system energy* of interpretation $\nu$ w.r.t. $\mathcal{M}_3$ is 30. Because we add the penalties of the formulae not satisfied, $\nu(p_i \rightarrow b_i)$ and $\nu(p_i \rightarrow \neg\, f_i)$, that is, 20 + 10.

In $\mathcal{M}_4$,
$\nu(p_i \rightarrow b_i) = -1$ because $\nu(p_i) = 1$ and $\nu(b_i) = -1$;
$\nu(b_i \rightarrow f_i) = 1$ because $\nu(b_i) = \nu(f_i) = -1$; and
$\nu(p_i \rightarrow \neg\, f_i) = 1$ because $\nu(p_i) = \nu(\neg\, f_i) = 1$.

Formulae satisfied in $\mathcal{M}_4$ are $\nu(b_i \rightarrow f_i)$ and $\nu(p_i \rightarrow \neg\, f_i)$. Model $\mathcal{M}_4$ can be thought of as a world in which birds fly but penguins do not. Note that in this world, penguins are not birds, $\nu(p_i \rightarrow \neg\, b_i) = 1$.

The *system energy* of interpretation $\nu$ w.r.t. $\mathcal{M}_4$ is 20. The formula not satisfied is $p_i \rightarrow b_i$ with penalty 20.

The *preferred* model w.r.t. to the set of models we have presented above, would be $\mathcal{M}_2$, because it has the minimal *system energy* of the interpretation $\nu$.

The models that we have presented above are the analogues of the scenarios in 4.2.1. For selecting the *optimal* scenario the minimal cost $\mathcal{K}$ is required. For choosing the *preferred* models, the minimal *system energy* of $\nu$ is needed. When we have parallel scenarios and models, the *optimal* scenario should coincide with the *preferred* model. In the example above, the *optimal* scenario was $\Delta_2$, its analogue in the semantics is model $\mathcal{M}_2$, this was the *preferred* model of $\nu$.

DEFINITION 4.8
Let $\alpha$, $\beta$ be two arbitrary formulae of a PKB $\Psi$.

$\alpha \models_\Psi \beta$ *if and only if* every preferred model of $\alpha$ is a model of $\beta$.

$\mathcal{M}_2$ was the *preferred* model w.r.t. the models of the PKB $\Psi$ presented above. From this model we could infer for example that $\mathcal{M}_2$, $p_i \models_\Psi b_i$ and $p_i \models_\Psi \neg f_i$. However, it does not follow that $\mathcal{M}_2$, $p_i \models_\Psi f_i$.

FACT 1. The logic exhibited above is sound and complete. The notions in definition 4.4 and 4.8 coincide.[7]

## 4.3 Translation of symmetric networks into penalty logic

In chapter 3 we introduced the function

$$E(s) = \quad -\sum_{i<j} w_{ij} s_i s_j$$

This function, called Ljapunov or *energy* function, permits us to calculate the value of E with respect to different activation states. When an activation state changes, E can decrease, increase or remain equal. In the previous chapter when we calculated E, we always pointed out the state with the *minimal energy* in the network. This state in the connectionist level was related to the output of an *optimal* linguistic candidate in the OT level. Now, we introduce another link, between states in a connectionist level and models in a penalty logic base. States with the *minimal energy* in a network will be linked to *preferred* models of the interpretation $\nu$.[8] In the example below we will see this link.

*Example*

Below we present an example of a translation of a local connectionist network into a penalty logic base. This translation not only will permit us to observe the link, states with *minimal energy* E and *preferred* models, but also to familiarize us with something that we will do in the next chapter.

In the example we use a local and symmetric connectionist network. We have selected this kind of network because is of the same type as $CV_{net}$, the network that we will translate in the next chapter. Remember that we are interested in translating the inborn constraints of OT, expressed in $CV_{net}$.

In chapter 3, we saw the adequacy of local and symmetric connectionist networks for representing OT constraints. Here, we want to point out that this kind of network is also adequate for representing or being represented by a penalty knowledge base. The translation from localist networks into penalty logic is relatively easy if we take in account that each node is represented by an atomic formula. In the other direction, it is also relatively simple, in the sense that each atomic propositional formula will represent a

[7] For a proof *cfr.* [19].

[8] The link of symmetric networks and penalty logic is treated in [19] and [4].

unit in the network. Weights in a network are analogous of penalties in a PKB. Weights can vary, but if so wished, they can be fixed into a symmetric network. In a PKB, penalties are easily fixed, we just need to assign the same cost or penalty to a set of formulae.

Notice that the following example even though it is quite similar to the example of the PKB *ex* presented in 4.1.1, it is different. In the PKB *ex* the formulae were conditionals. Below the formulae are biconditionals because in this way we can translate the symmetric character of the weights in the network.

We will start calculating the energy E for the network. Then, we translate the network into a PKB. After that, we are able to draw the link states with *minimal energy* E in a network and the *preferred* models of the interpretation $\nu$.

Consider the following network:

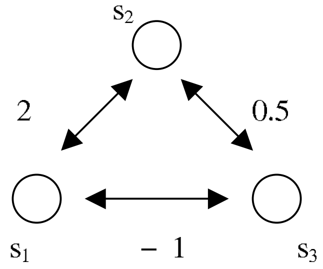


Figure 4.1: A symmetric network

Take a set of activation states $\mathcal{S} = [-1,\, 1]$. The energy function for this network is:

$$\mathrm{E}(s) = -\ 2\ s_1 s_2\ -\ 0.5\ s_2\ s_3\ +\ s_1 s_3$$

Let us consider only the states in which $s_1$ is activated, i.e. when $s_1$ has value 1.

The state $s$ that shows the minimal energy is $\langle 1, 1, -1\rangle$ with $\mathrm{E} = -2.5$.

Now, we translate the network into a PKB. To obtain the translation we follow the next three steps:

- Each node of the network is represented by an atomic propositional formula.
- Each symmetric weight by a biconditional.
- Negative weights are represented by negations.

| | $s$ | | | E |
|---|---|---|---|---|
| | $s_1$ | $s_2$ | $s_3$ | |
| [1] | 1 | 1 | 1 | $-1.5$ |
| [2] | 1 | 1 | $-1$ | $-2.5$ |
| [3] | 1 | 1 | 0 | $-2$ |
| [4] | 1 | $-1$ | 1 | 3.5 |
| [5] | 1 | $-1$ | $-1$ | 0.5 |
| [6] | 1 | $-1$ | 0 | 2 |
| [7] | 1 | 0 | 1 | 1 |
| [8] | 1 | 0 | $-1$ | $-1$ |
| [9] | 1 | 0 | 0 | 0 |

Table 4.2: E for activated states $s_1$

Take $p_i$, $b_i$, $f_i$ to represent respectively the nodes $s_1$, $s_2$, $s_3$. The PKB of the network in figure 4.1 is:

| | |
|---|---|
| $p_i \leftrightarrow b_i$ | 2 |
| $b_i \leftrightarrow f_i$ | 0.5 |
| $p_i \leftrightarrow \neg f_i$ | 1 |

Define $\nu(\varphi \leftrightarrow \psi) = 1$ *iff* $\nu(\varphi) = \nu(\psi)$. Otherwise, $\nu(\varphi \leftrightarrow \psi) = -1$. Consider the models in which $\nu(p_i) = 1$:

| | | | | *penalties* | | | |
|---|---|---|---|---|---|---|---|
| | | | | 2 | 0.5 | 1 | |
| | $p_i$ | $b_i$ | $f_i$ | $p_i \leftrightarrow b_i$ | $b_i \leftrightarrow f_i$ | $p_i \leftrightarrow \neg f_i$ | $\mathcal{E}_{ex}(\nu)$ |
| $\mathcal{M}'_1$ | 1 | 1 | 1 | 1 | 1 | $-1$ | 1 |
| $\mathcal{M}'_2$ | 1 | 1 | $-1$ | 1 | $- 1$ | 1 | 0.5 |
| $\mathcal{M}'_3$ | 1 | $-1$ | 1 | $-1$ | $-1$ | $-1$ | 3.5 |
| $\mathcal{M}'_4$ | 1 | $-1$ | $-1$ | $-1$ | 1 | 1 | 2 |

Table 4.3: Some models $\mathcal{M}$

The model with the *minimal energy* of $\nu$ is $\mathcal{M}'_2$ with 0.5, so, $\mathcal{M}'_2$ is the *preferred* model.

Model $\mathcal{M}'_2$ is the analogon of the state [2] shown in table 4.2. Observe that in $\mathcal{M}'_2$ the propositions $p_i$, $b_i$, $f_i$ take respectively the values 1, 1, $-1$, i.e. the first two formulae are satisfied while the third one is not. These values correspond to the state [2] in table 4.2 because the states $s_1$, $s_2$, $s_3$

have respectively the values 1, 1, $-1$, i.e. the first two nodes are activated while the third one is not. So, we can say that the *preferred* model $\mathcal{M}$'$_2$, i.e. the model with the minimal *system energy* of $\nu$, corresponds to the state $s$ with the *minimal energy* E in the network.

Note that the interpretation $\nu$, as we have defined it, is bivalent. We are considering only the cases in which a proposition gets a value 1 or $-1$, that is, when a formula is satisfied or not. We are not including intermediate cases. We are pointing this out because for calculating the energy E of the network we used a rest state indicated by zero. These states are not going to be considered for mapping in the interpretation $\nu$. In other words, we take from the table 4.2 just the states in [1], [2], [4] and [5]. Notice that we do not map the zeros of the the E-calculation to $-1$ in the $\nu$-interpretation. If we do this, we cannot state the order among the states in the network. This is important because we need to take the state that presents the *minimal energy* E, in order to compare it with the *preferred* model.

Consider the states $\langle 1, 1, -1\rangle$ and $\langle 1, 1, 0\rangle$. If we assume the mapping from zero of E to $-1$ of $\nu$, those states will be mapped to the same model {1, 1, 0}. The problem with this arises when we compare the *preferred* model with the state with *minimal* energy. If the *preferred* model is {1, 1, 0 }, this model will correspond to ambiguous energy E, specifically with E $= -$ 2.5 and 2. In other words, the *preferred* model will not necessarily be linked to the state with minimal energy.

A problem that can arise by the exclusion of the mapping of the zeros lies in the following question: what happens if the state of minimal energy contains a zero? How can it be restored by the PKB? One way would be to add a third value in the interpretation $\nu$, i.e. to transform it to trivalency. In this case, we would need to redefine the connectives in such way that the order of the states can be preserved in the models.

In this thesis we are going to stick to the bivalent case, leaving the trivalent one for further research. Taking this into account, we are going to consider only the bivalent states in the network, that is, states that just are activate or inactivate nodes (none zero). From this set of states, the state(s) with the *minimal energy* E will be the analogon of the *preferred* model(s) under the interpretation $\nu$ as defined in bivalent terms. In other words, we state a link between networks and penalty logic under the bivalency assumption. This assumption seems to be enough for restoring the cases of *optimal* candidates in OT.

## Chapter 5

# Translation of CV$_{net}$ into Penalty Logic

According to the nativist position in OT, the universals of well-formedness $\mathcal{U}$ are encoded in the genome. The genome equips the learner with a Language Acquisition Device (LAD), here called $CV_{net}$. Our proposal is that such a genomic encoding can be expressed in a simpler way in the language of penalty logic. In particular, we suggest that the penalty knowledge base *cv* becomes part of the LAD that encodes the knowledge of $\mathcal{U}$.

In order to form the PKB *cv* we translate the constraints of $CV_{net}$ into penalty logic. Then, we assign some abstract penalties to the constraints translated. After this, we draw the link of *minimal energy* E-*preferred* model between the networks of the $n$-constraints presented in chapter 3 and the *preferred* models of the PKB *cv*. After having done this, we link these notions with the *optimal* candidate notion in OT. For this link we first look at which models would be the *preferred* ones under single constraints and then, we recall an OT table presented in chapter 2 that contains the *Con* constraints in order to see how the *optimal* candidate in OT can be preserved in the other levels.

### 5.1 Formulae

In $CV_{net}$ we distinguished units by being input, output or correspondence units, or by being vowel or consonant units. We want to keep these distinctions in our symbolism. For keeping the first distinction we use single-upper indexes for input units, single-down indexes for output units, and double-down indexes for correspondence units. For the other distinction we use $v$ for vowels, $c$ for consonants, $co$ for onset consonants and $cc$ for coda consonants.

Let INPUT be the set of input units, OUTPUT the set of output units, and CORRESP the set of correspondence units. Let $i$, $j$ be indexes that can stand for any natural number. We have the following sets of propositional atoms:

A set { $v^0$, $v^1$, $\ldots$, $v^n$ } where $v^i$ stands for: 'unit $i$ is a vowel input unit'.
A set { $c^0$, $c^1$, $\ldots$, $c^n$ } where $c^i$ stands for: 'unit $i$ is a consonant input unit'.
{ $v^0$, $v^1$, ..., $v^n$ } and { $c^0$, $c^1$, ..., $c^n$ } belong to INPUT.

A set $\{v_{ij}\}$ where $i \geq 0$, $j \geq 0$, and $v_{ij}$ stands for unit $i$ from $j$ is a correspondence vowel unit.
A set $\{c_{ij}\}$ where $i \geq 0$, $j \geq 0$, and $c_{ij}$ stands for unit $i$ from $j$ is a correspondence consonant unit.
$\{v_{ij}\}$ and $\{c_{ij}\}$ belong to CORRESP.

A set { $v_0$, $v_1$, $\ldots$, $v_n$ } where $v_j$ stands for: 'unit $j$ is a vowel output unit'.
A set { $co_0$, $co_1$, $\ldots$, $co_n$ } where $co_j$ stands for: 'unit $j$ is a onset (output) unit'.
A set { $cc_0$, $cc_1$, $\ldots$, $cc_n$ } where $cc_j$ stands for: 'unit $j$ is a coda (output) unit'.
The sets { $v_0$, $v_1$, $\ldots$, $v_n$ }, { $co_0$, $co_1$, $\ldots$, $co_n$ } and { $cc_0$, $cc_1$, $\ldots$, $cc_n$ } belong to OUTPUT.

A set { $b_0$, $b_1$, $\ldots$, $b_n$ } where $b_i$ stands for: 'unit $i$ is a bias'.

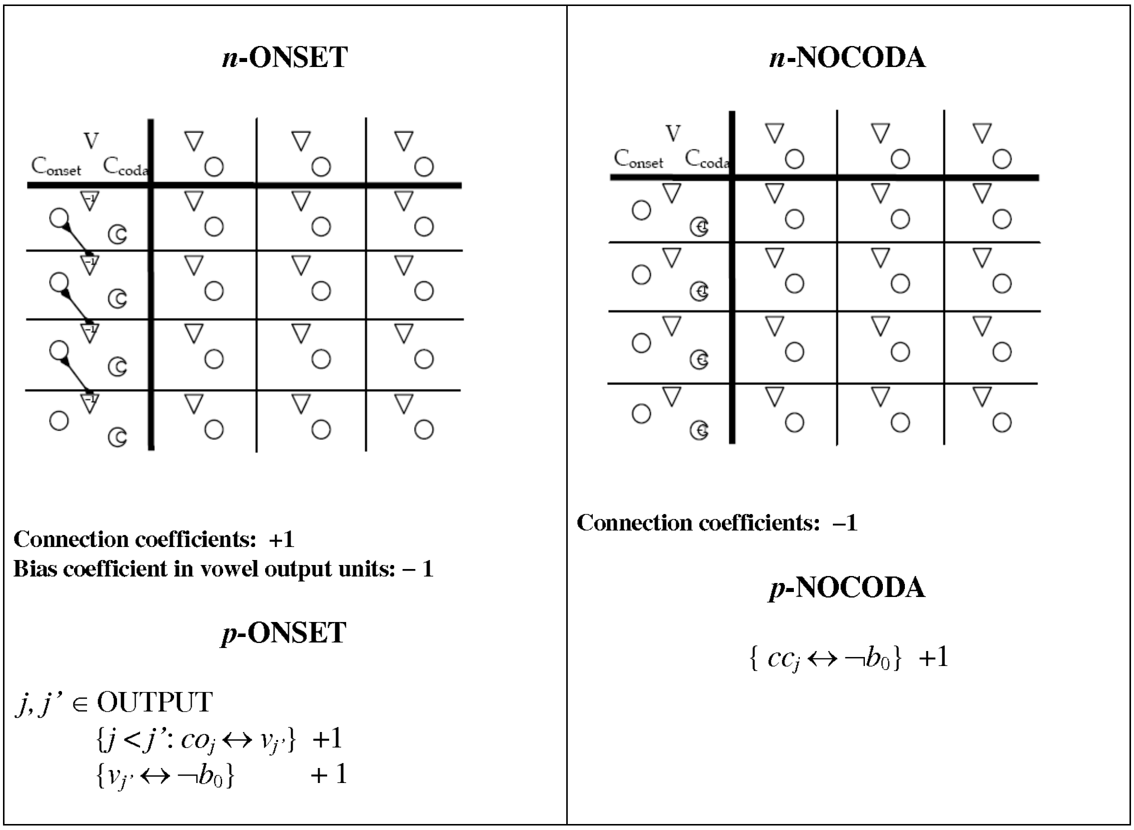
## 5.2 Translation of $CV_{net}$

### 5.2.1 Formulae of *cv*

We translate the following $n$-constraints: All the *net-Con* constraints, *net-Identity* and *net-Linearity* from the *net-Gen* set; *net-OutputIdentity* and *net-Correspondence* from the *net-Structural* set. Constraints in the version of Penalty Logic will be called $p$-constraints. Below we repeat the networks of some *net*-constraints introduced in chapter 3 and next to each one, we write down the set of formulae of the corresponding $p$-constraints.

We have translated the negative connection coefficients and biases as negations. For example, see *net-Parse* that assumes bias coefficient $-1$ in input units. The negative value is represented in the negation of the bias: { $v^i \leftrightarrow \neg\, b_0$ }.

Additionally, you will see numbers (with a '+' sign) next to each formula. These numbers indicate the connection coefficient and biases that each constraint had in $CV_{net}$, they are not the penalties. Penalties in the logical language represent the weights of a network, but in the cases below penalties cannot be assigned properly because weigths are not specified.[1] In the next section, we will assume certain penalties for the $p$-constraints in order to explain how the PKB *cv* might work.



***n*-ONSET**

Connection coefficients: +1
Bias coefficient in vowel output units: – 1

***p*-ONSET**

$j, j' \in$ OUTPUT

$\{j < j' : co_j \leftrightarrow v_{j'}\}$ +1

$\{v_{j'} \leftrightarrow \neg b_0\}$ + 1

***n*-NOCODA**

Connection coefficients: –1

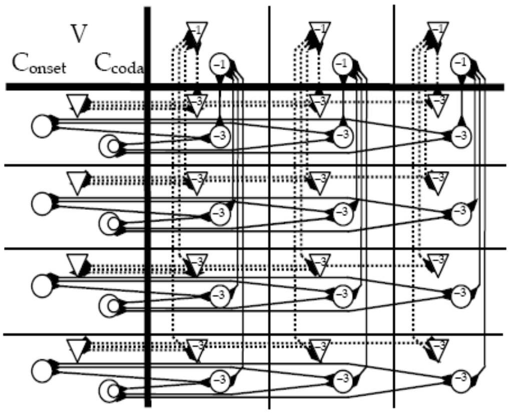
***p*-NOCODA**

$\{ cc_j \leftrightarrow \neg b_0\}$ +1

Figure 5.1: *p-Onset* and *p-NoCoda*

[1] The networks presented in chapter 3 and repeated below are generalizations of how microconstraints can be implemented, assuming certain connection coefficients and in some cases certain biases, but they do not show activation that indicate to us the possible strengths between two nodes, i.e. the specific weights.

***n*-PARSE**



**Connection coefficients: +2**
**Bias coefficients in input units: −1**
**Bias coefficients in correspondence units: −3**

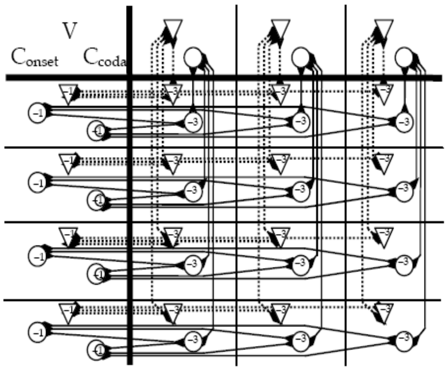
***p*-PARSE**

$\{v^i \leftrightarrow v_{ij}\}$ +2
$\{v_j \leftrightarrow v_{ij}\}$ +2
$\{v^i \leftrightarrow \neg\, b_0\}$ +1
$\{v_{ij} \leftrightarrow \neg b_1\}$ +3

$\{c^i \leftrightarrow c_{ij}\}$ +2
$\{co_j \leftrightarrow c_{ij}\}$ +2
$\{cc_j \leftrightarrow c_{ij}\}$ +2
$\{\ c^i \leftrightarrow \neg\, b_2\}$ +1
$\{c_{ij} \leftrightarrow \neg b_3\}$ +3

Figure 5.2: *p-Parse*

***n*-FILL$^{V}$ (dotted lines)**
***n*-FILL$^{C}$ (solid lines)**



**Connection coefficients: +2**
**Bias coefficients in output units: –1**
**Bias coefficients in correspondence units: –3**

***p*- FILL$^{V}$**

$\{v^{i} \leftrightarrow v_{ij}\}$ +2
$\{v_{j} \leftrightarrow v_{ij}\}$ +2
$\{v_{ij} \leftrightarrow \neg b_{0}\}$ +3
$\{v_{j} \leftrightarrow \neg b_{1}\}$ +1

***p*- FILL$^{C}$**

$\{c^{i} \leftrightarrow c_{ij}\}$ +2
$\{co_{j} \leftrightarrow c_{ij}\}$ +2
$\{cc_{j} \leftrightarrow c_{ij}\}$ +2
$\{c_{ij} \leftrightarrow \neg b_{2}\}$ +3
$\{co_{j} \leftrightarrow \neg b_{3}\}$ +1
$\{cc_{j} \leftrightarrow \neg b_{4}\}$ +1

Figure 5.3: *p-Fill*$^{V}$ and *p-Fill*$^{C}$

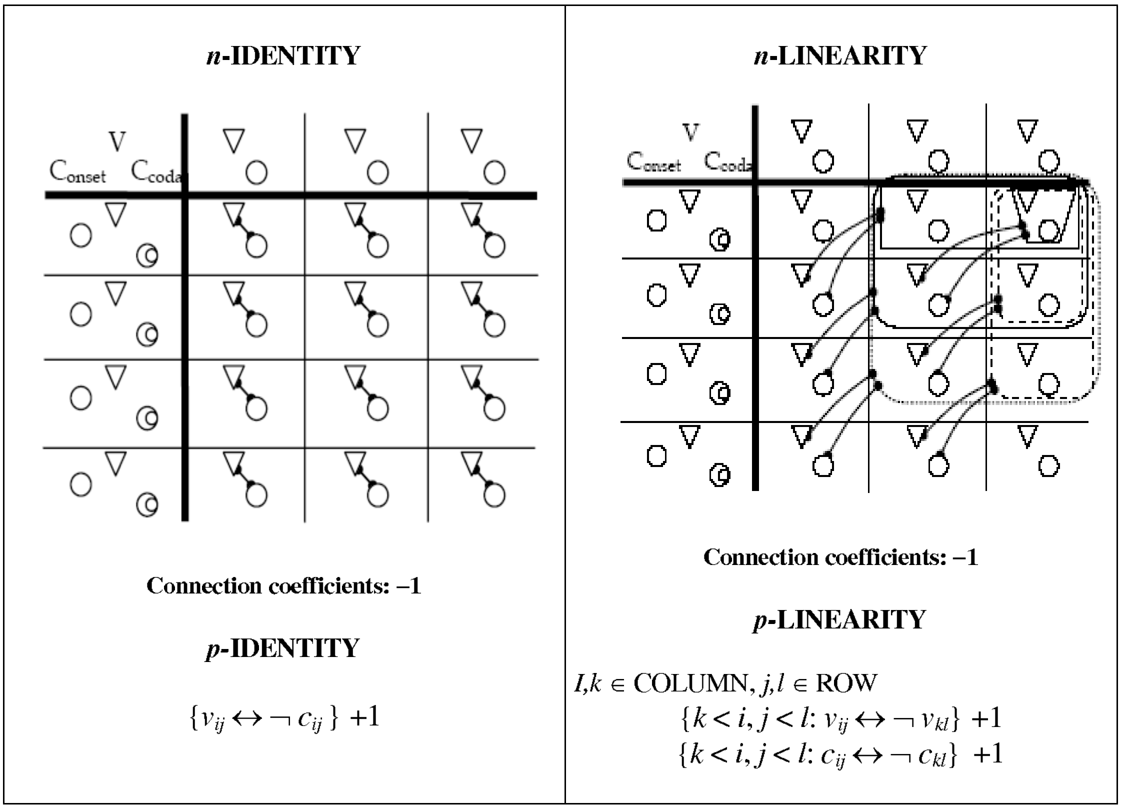
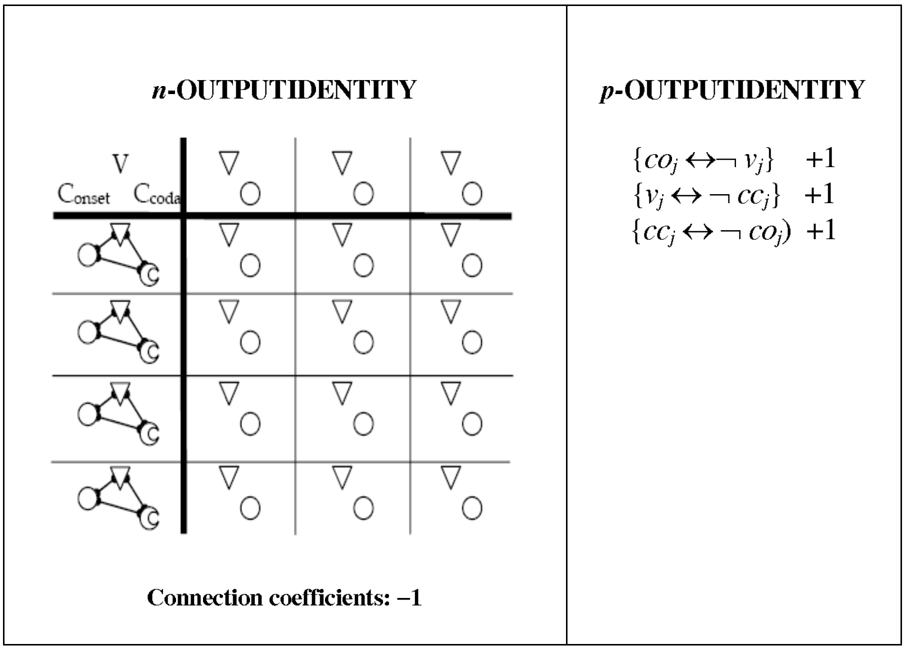

n-IDENTITY
V
Conset
Ccoda
Connection coefficients: –1
p-IDENTITY
$\{v_{ij} \leftrightarrow \neg c_{ij}\}$ +1
n-LINEARITY
V
Conset
Ccoda
Connection coefficients: –1
p-LINEARITY
I,k ∈ COLUMN, j,l ∈ ROW
$\{k < i, j < l: v_{ij} \leftrightarrow \neg v_{kl}\}$ +1
$\{k < i, j < l: c_{ij} \leftrightarrow \neg c_{kl}\}$ +1

Figure 5.4: *p-Identity*



$$\{co_j \leftrightarrow \neg\, v_j\} \quad +1$$
$$\{v_j \leftrightarrow \neg\, cc_j\} \quad +1$$
$$\{cc_j \leftrightarrow \neg\, co_j) \quad +1$$

Figure 5.5: *p-OutputIdentity*

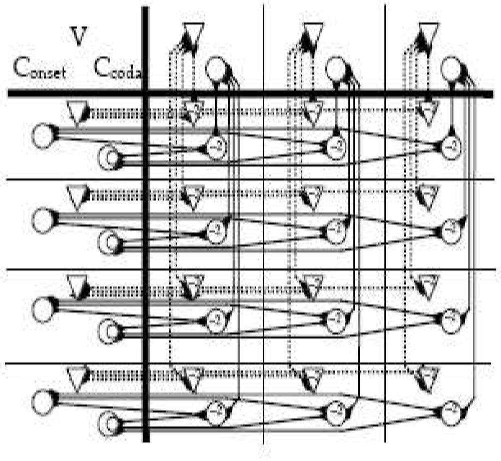
### 5.2.2 Penalties of *cv*

A PKB includes logical formulae plus their corresponding penalties. The formulae have been presented in the previous section. Note that we have used a set of formulae instead of a set of sentences in the translation of the $n$-constraints. This level of abstraction will permit cases in which a $p$-constraint is satisfied and cases in which it is not. We are interested in the cases in which a $p$-constraint shows the highest number of satisfied formulae. That is, in the *preferred* models of each $p$-constraint. But in order to determine such models we will need to assign specific penalties to the $p$-constraints. In this section, we introduce the penalties in an abstract way, leaving a case containing more concrete penalties for the following section.

The *p-Gen* and the *p-Structural* constraints are the inborn non- hierarchizable constraints. We assume that penalties for these constraints are higher than any penalty of a *net-Con* constraint. Besides, we consider the penalties of the *p-Gen* and the *p-Structural* fixed, analogous to the fixed weights of the *net-Gen* and *net-Structural* constraints. Penalties for the *p-Gen* and the *p-Structural* will be indicated by an $n^*$, where $n^*$ can be substituted by a real positive number larger than any number assigned to a *p-Con* constraint.

The *p-Con* constraints can vary their penalty according to their position in a hierarchy. The hierarchy on $p$-constraints will be kept through

***n*-CORRESPONDENCE**



**Connection coefficients: +1**
**Bias coefficients in correspondence units: –2**

***p*-CORRESPONDENCE**

$\{v^i \leftrightarrow v_{ij}\}$ +1
$\{v_j \leftrightarrow v_{ij}\}$ +1
$\{v_{ij} \leftrightarrow \neg b_0\}$ +2

$\{c^i \leftrightarrow c_{ij}\}$ +1
$\{co_j \leftrightarrow c_{ij}\}$ +1
$\{cc_j \leftrightarrow c_{ij}\}$ +1
$\{c_{ij} \leftrightarrow \neg b_1\}$ +2

Figure 5.6: *p-Correspondence*

exponential penalties, with an order analogous to exponential weights. The $p$-constraint's penalties or non-fixed penalties will be indicated by $(\frac{v}{n})^k$, where $v$ represents the number of violations, $n$ is a natural number representing the total number of violations possible for a single $p$-constraint and $k$ a natural number that indicates the position of the $p$-constraint in the hierarchy.

Below we present a version of a PKB *cv*. Note that this PKB is incomplete, given the missing formulae of some $p$-constraints.

## 5.3 The *preferred* models

For simplicity, in what follows we only consider the rankeable constraints, i.e. the *p-Con* constraints. The *p-Gen* and the *p-Structural* constraints are implicitly assumed having the highest penalty of all *p-Con* constraints.

In the OT level, candidates were evaluated with respect to constraints that had a position in a hierarchy. The winner candidate was always the *optimal*. Analogous to this evaluation, in Penalty Logic we will consider a hierarchy on $p$-constraints. This hierarchy will be built with exponential penalties, and from this will emerge the analogue of an *optimal* candidate, that is a *preferred* model.

In this section, we focus on the selection of an 'optimal' model under a single *p-Con* constraint, leaving the selection under a hierarchy for a later section.

Consider the *p-Onset* constraint. The corresponding models are shown in table 5.2.

Independently of which particular penalty the biconditionals get, the models that come out as the *preferred* models are in rows 2 and 7. These models show all the formulae of *p-Onset* satisfied, that is, $\nu(co_j \leftrightarrow v_{j'}) = \nu(v_{j'} \leftrightarrow \neg b_0) = 1$ (see colums 5, 6 in table 5.2 below).

Looking at the propositional atoms, a model of *p-Onset* has a minimal *system energy* of $\nu$ in two cases: a) when the formulae expressing that there is an onset and a vowel are both satisfied, but not the formula expressing there is a bias; and b) when the formulae about onset and vowel are both not satisfied but the formula of the bias is.

The *preferred* models of *p-Onset* match with the states showing minimal energy E in the connectionist level. As a particular case, you can compare table 5.2 with table 3.1 introduced in chapter 3. The minimal energy E obtained in the system arose in the analogous cases of the *preferred* models just mentioned: when an onset and a vowel unit were both activated but the bias was not, and when both, an onset and a vowel unit were disactivated, but the bias was activated.

For the *p-NoCoda* constraint the models are shown in table 5.3.

| | *p-constraints* | *formulae* | *penalty* |
|---|---|---|---|
| 1 | *p-Onset* | $\{j < j' : co_j \leftrightarrow v_{j'}\}$<br>$\{v_{j'} \leftrightarrow \neg b_0\}$ | $(\frac{v}{n})^k$ |
| 2 | *p-NoCoda* | $\{cc_j \leftrightarrow \neg b_0\}$ | $(\frac{v}{n})^k$ |
| 3 | *p-Parse* | $\{v^i \leftrightarrow v_{ij}\}$<br>$\{v_j \leftrightarrow v_{ij}\}$<br>$\{v^i \leftrightarrow \neg b_0\}$<br>$\{v_{ij} \leftrightarrow \neg b_1\}$<br>$\{c^i \leftrightarrow c_{ij}\}$<br>$\{co_j \leftrightarrow c_{ij}\}$<br>$\{cc_j \leftrightarrow c_{ij}\}$<br>$\{c^i \leftrightarrow \neg b_2\}$<br>$\{c_{ij} \leftrightarrow \neg b_3\}$ | $(\frac{v}{n})^k$ |
| 4 | *p-Fill*$^V$ | $\{v^i \leftrightarrow v_{ij}\}$<br>$\{v_j \leftrightarrow v_{ij}\}$<br>$\{v_{ij} \leftrightarrow \neg b_0\}$<br>$\{v_j \leftrightarrow \neg b_1\}$ | $(\frac{v}{n})^k$ |
| 5 | *p-Fill*$^C$ | $\{c^i \leftrightarrow c_{ij}\}$<br>$\{co_j \leftrightarrow c_{ij}\}$<br>$\{cc_j \leftrightarrow c_{ij}\}$<br>$\{c_{ij} \leftrightarrow \neg b_2\}$<br>$\{co_j \leftrightarrow \neg b_3\}$<br>$\{cc_j \leftrightarrow \neg b_4\}$ | |
| 6 | *p-Identity* | $\{v_{ij} \leftrightarrow \neg c_{ij}\}$ | $n*$ |
| 7 | *p-Linearity* | $k < i, j < l : v_{ij} \leftrightarrow \neg v_{kl}\}$<br>$k < i, j < l : c_{ij} \leftrightarrow \neg c_{kl}\}$ | $n*$ |
| 8 | *p-Integrity* | | $n*$ |
| 9 | *p-Uniformity* | | $n*$ |
| 10 | *p-OutputIdentity* | $\{co_j \leftrightarrow \neg v_j\}$<br>$\{v_j \leftrightarrow \neg cc_j\}$<br>$\{cc_j \leftrightarrow \neg co_j\}$ | $n*$ |
| 11 | *p-NoOutputGaps* | | $n*$ |
| 12 | *p-Correspondence* | $\{v^i \leftrightarrow v_{ij}\}$<br>$\{v_j \leftrightarrow v_{ij}\}$<br>$\{v_{ij} \leftrightarrow \neg b_0\}$<br>$\{c^i \leftrightarrow c_{ij}\}$<br>$\{co_j \leftrightarrow c_{ij}\}$<br>$\{cc_j \leftrightarrow c_{ij}\}$<br>$\{c_{ij} \leftrightarrow \neg b_1\}$ | $n*$ |
| 13 | *p-Nucleus* | | $n*$ |

Table 5.1: The PKB *cv*

|  | $co_i$ | $v_{j'}$ | $b_0$ | $co_j \leftrightarrow v_{j'}$ | $v_{j'} \leftrightarrow \neg b_0$ | $\neg b_0$ | $\mathcal{E}(\nu)$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | −1 | −1 |  |
| 2 | 1 | 1 | − 1 | **1** | **1** | 1 | **0** |
| 3 | 1 | −1 | 1 | −1 | 1 | −1 |  |
| 4 | 1 | −1 | −1 | −1 | −1 | 1 |  |
| 5 | −1 | 1 | 1 | −1 | −1 | −1 |  |
| 6 | −1 | 1 | −1 | − 1 | 1 | 1 |  |
| 7 | −1 | −1 | 1 | **1** | **1** | −1 | **0** |
| 8 | −1 | −1 | −1 | 1 | −1 | 1 |  |

Table 5.2: *p-Onset* models

|  | $cc_j$ | $b_0$ | $cc_j \leftrightarrow \neg b_0$ | $\neg b_0$ | $\mathcal{E}(\nu)$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | −1 | −1 |  |
| 2 | 1 | − 1 | 1 | 1 | **0** |
| 3 | −1 | 1 | 1 | −1 | **0** |
| 4 | −1 | −1 | −1 | 1 |  |

Table 5.3: *p-NoCoda* models

The *preferred* models are those in rows 2 and 3 (table 5.3). Observe that they are the unique models that have the formula constituting *NoCoda* satisfied, that is, $\nu(cc_j \leftrightarrow \neg b_0) = 1$. The *preferred* models of *p-NoCoda* coincide with the states showing the *minimal energy* E. The states $\langle 1, -1 \rangle$ and $\langle -1, 1 \rangle$ in table 3.2 have minimal energy E.

In the OT level the *Parse* constraint is satisfied if for every vowel or consonant in the input there is a correspondent vowel or (onset or coda) consonant in the output. In the case of *p-Parse*, there are two cases in which we have *preferred* models. When all the atomic formulae of *p-Parse* representing the content in the input, correspondence and output units are satisfied or when they are not (see rows 4 and 29 table 5.4). As you can see in table 5.4, when the formulae representing input, correspondence and output units are satisfied the biases are not (row 4); and, when such formulae are unsatisfied, biases are satisfied (row 29). This resembles the cases of *minimal energy* E of *Parse*. If you see table 3.3 (chapter 3), the states in rows [4] and and [29] show the *minimal energy* E with −8. In [4] the input, correspondence and output units are all activated but the biases are not, whereas in [29] the reverse is the case.

|  | $v^i$ | $v_{ij}$ | $v_j$ | $b_0$ | $b_1$ | $v^i \leftrightarrow v_{ij}$ | $v_j \leftrightarrow v_{ij}$ | $v^i \leftrightarrow \neg b_0$ | $\neg b_0$ | $v_{ij} \leftrightarrow \neg b_1$ | $\neg b_1$ | $\mathcal{E}(\nu)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | −1 | −1 | −1 | −1 |  |
| 2 | 1 | 1 | 1 | 1 | −1 | 1 | 1 | −1 | −1 | 1 | 1 |  |
| 3 | 1 | 1 | 1 | −1 | 1 | 1 | 1 | 1 | 1 | −1 | −1 |  |
| 4 | 1 | 1 | 1 | −1 | −1 | **1** | **1** | **1** | 1 | **1** | 1 | **0** |
| 5 | 1 | 1 | −1 | 1 | 1 | 1 | −1 | −1 | −1 | −1 | −1 |  |
| 6 | 1 | 1 | −1 | 1 | −1 | 1 | −1 | −1 | −1 | 1 | 1 |  |
| 7 | 1 | 1 | −1 | −1 | 1 | 1 | −1 | 1 | 1 | −1 | −1 |  |
| 8 | 1 | 1 | −1 | −1 | −1 | 1 | −1 | 1 | 1 | 1 | 1 |  |
| 9 | 1 | −1 | 1 | 1 | 1 | −1 | −1 | −1 | −1 | 1 | −1 |  |
| 10 | 1 | −1 | 1 | 1 | −1 | −1 | −1 | −1 | −1 | −1 | 1 |  |
| 11 | 1 | −1 | 1 | −1 | 1 | −1 | −1 | 1 | 1 | 1 | −1 |  |
| 12 | 1 | −1 | 1 | −1 | −1 | −1 | −1 | 1 | 1 | −1 | 1 |  |
| 13 | 1 | −1 | −1 | 1 | 1 | −1 | 1 | −1 | −1 | 1 | −1 |  |
| 14 | 1 | −1 | −1 | 1 | −1 | −1 | 1 | −1 | −1 | −1 | 1 |  |
| 15 | 1 | −1 | −1 | −1 | 1 | −1 | 1 | 1 | 1 | 1 | −1 |  |
| 16 | 1 | −1 | −1 | −1 | −1 | −1 | 1 | 1 | 1 | −1 | 1 |  |
| 17 | −1 | 1 | 1 | 1 | 1 | −1 | 1 | 1 | −1 | −1 | −1 |  |
| 18 | −1 | 1 | 1 | 1 | −1 | −1 | 1 | 1 | −1 | 1 | 1 |  |
| 19 | −1 | 1 | 1 | −1 | 1 | −1 | 1 | −1 | 1 | −1 | −1 |  |
| 20 | −1 | 1 | 1 | −1 | −1 | −1 | 1 | −1 | 1 | 1 | 1 |  |
| 21 | −1 | 1 | −1 | 1 | 1 | −1 | −1 | 1 | −1 | −1 | −1 |  |
| 22 | −1 | 1 | −1 | 1 | −1 | −1 | −1 | 1 | −1 | 1 | 1 |  |
| 23 | −1 | 1 | −1 | −1 | 1 | −1 | −1 | −1 | 1 | −1 | −1 |  |
| 24 | −1 | 1 | −1 | −1 | −1 | −1 | −1 | −1 | 1 | 1 | 1 |  |
| 25 | −1 | −1 | 1 | 1 | 1 | 1 | −1 | 1 | −1 | 1 | −1 |  |
| 26 | −1 | −1 | 1 | 1 | −1 | 1 | −1 | 1 | −1 | −1 | 1 |  |
| 27 | −1 | −1 | 1 | −1 | 1 | 1 | −1 | −1 | 1 | 1 | −1 |  |
| 28 | −1 | −1 | 1 | −1 | −1 | 1 | −1 | −1 | 1 | −1 | 1 |  |
| 29 | −1 | −1 | −1 | 1 | 1 | **1** | **1** | **1** | −1 | **1** | −1 | **0** |
| 30 | −1 | −1 | −1 | 1 | −1 | 1 | 1 | 1 | −1 | −1 | 1 |  |
| 31 | −1 | −1 | −1 | 1 | −1 | 1 | 1 | 1 | −1 | −1 | 1 |  |
| 32 | −1 | −1 | −1 | −1 | −1 | 1 | 1 | −1 | 1 | −1 | 1 |  |

Table 5.4: *p-Parse* models

Observe that in the table above we have considered the cases in which *p-Parse* has vowels as inputs. Cases in which a consonant should be mapped into the output are quite similar to the case already presented. In such cases, the same *preferred* models arise, with the difference that for consonants we would have to consider the set of formulae $\{ c^i \leftrightarrow c_{ij}, cc_j \leftrightarrow c_{ij}, c^i \leftrightarrow \neg b_2, c_{ij} \leftrightarrow \neg b_3 \}$ depending on whether the consonant is mapped to an onset or coda consonant.

In the case of *p-Fill*$^V$ we have a similar table to *p-Parse*, only we have to substitute the formulae that indicates bias in the input for the formulae expressing bias in the output. In general terms, the structure of the formulae of this constraint compared with *p-Parse* is similar. The *preferred* models are those in which all formulae of *p-Fill*$^V$ are satisfied. That is, in the cases where the atomic formulae show the values: $\nu(v^i) = \nu(v_{ij}) = \nu(v_j) = 1$ and $\nu(b_0) = \nu(b_1) = -1$, or $\nu(v^i) = \nu(v_{ij}) = \nu(v_j) = -1$ and $\nu(b_0) = \nu(b_1) = 1$.

A case analogous for *p-Fill*$^C$, where the preferred models are the same valuations as in *p-Fill*$^V$. The difference is that in the case of *p-Fill*$^C$ we have to consider the corresponding formulae with consonants instead of those with vowels that constitute *p-Fill*$^V$.

## 5.4 Exponential penalties

The *preferred* models of the *p*-constraints shown above are analogues of the *optimal* candidates under a single constraint in the OT level. In this section we recall an OT table presented in chapter 2, and we explain how the hierarchy of constraints can be represented at the Penalty Logic level. What we explain in this section together with what saw in the previous section will permit us to close the link *minimal energy* E$-$*preferred* models$-$*optimal* candidates.

For an exemplification of how an OT table can be represented in a PKB, we need certain exponential penalties. This kind of penalty will introduce a hierarchy among the constraints. For the example below, we will use a decimal position system. More explicitly, we will assume a system that takes the $n$ in $(\frac{v}{n})^k$ equal to 10. But as we have said, $n$ is a natural number that represents the number of violations that are possible under a single constraint. This means that $n$ might be infinite.

In Penalty Logic the hierarchy can be introduced as follows. The first constraint in the OT hierarchy from top to down is assigned with a penalty $p$, where $p$ can be any natural number in the range of 1 to 10 that represents the number of violations of the highest OT-constraint. The remaining constraints going down in the OT hierarchy are assigned a penalty $(\frac{v}{n})^k$, where $v$ represents the number of violations (or asterisks in OT table) and $k$ is a natural number that grows as the hierarchy goes down, i.e. $(\frac{v}{n})^1$,$(\frac{v}{n})^2$,$\ldots(\frac{v}{n})^n$. For example, in table 5.5 the first constraint after the highest one is *Parse*,

| input: $/\mathrm{V}^1\mathrm{C}^2\mathrm{V}^3\mathrm{C}^4/ \rightarrow$ | $\mathit{Fill}^C \gg \mathit{Parse} \gg \mathit{Fill}^V \gg \mathit{NoCoda} \gg \mathit{Onset}$ | | | | |
|---|---|---|---|---|---|
| $\mathrm{c}_1$: $.\square\mathrm{V}^1.\mathrm{C}^2\mathrm{V}^3.\langle C\rangle$ | * | * | | | |
| $\mathrm{c}_2$: $\langle V\rangle.\mathrm{C}^2\mathrm{V}^3.\langle C\rangle$ | | ** | | | |
| $\mathrm{c}_3$: $\langle V\rangle.\mathrm{C}^2\mathrm{V}^3.\mathrm{C}^4\square.$ | | * | * | | |
| $\mathrm{c}_4$: $.\mathrm{V}^1.\mathrm{C}^2\mathrm{V}^3\mathrm{C}^4.$ | | | | * | * |

Table 5.5: OT table

for it $k = 1$ is chosen, for $\mathit{Fill}^V$, $k$=2, and so on. Notice that this way of assigning penalties only accepts a maximum of 9 violations per constraint, given the decimal character we have assumed.

The positional location of the violation will permit us to join in a single string the violations of constraints for a certain candidate. In that way, we can assign a single complex penalty for each output candidate instead of one simple penalty per constraint.

Recall formula $F3$ in chapter 4. This formula permits us to calculate the *system energy* of the interpretation $\nu$. According to that formula we should add up the penalties of the formulae that belong to the corresponding PKB but are not satisfied. The violations on OT candidates marked by an asterisk are the analogues of non-satisfied formulae in Penalty Logic, that is, formulae with value $-1$. If we consider the table 5.5, in a model $\mathcal{M}_1$ that resembles candidate $\mathrm{c}_1$, the formulae not satisfied would be those of *p*-$\mathit{Fill}^C$ and *p-Parse*, or in other terms, model $\mathcal{M}_1$ would not be a *preferred* model, neither under *p*-$\mathit{Fill}^C$ nor under *p-Parse*. Adding up the penalties of these two *p*-constraints, that is 1 + 0.1, we get 1.1. This number represents the *system energy* of the interpretation $\nu$ under the hierarchy H. The energy of the rest of models can be calculated in a similar way.

The *system energy* of $\nu$ with respect to the candidate-models in the table above is:
Candidate $\mathrm{c}_1$: penalty of $\mathit{Fill}^C$ 1 + *Parse* 0.1 = **1.1**
Candidate $\mathrm{c}_2$: penalty of *Parse* 0.1 × 2 = **0.2**
Candidate $\mathrm{c}_3$: penalty of *Parse* 0.1 + $\mathit{Fill}^V$ 0.01 = **0.11**
Candidate $\mathrm{c}_4$: penalty of *NoCoda* 0.001 + *Onset* 0.0001 = **0.0011**

The candidate with the minimal *system energy* is the *preferred* model. In this case candidate $\mathrm{c}_4$ would be that model. Observe that this candidate has penalty 0.0011, that is, the minimal one because $0.0011 < 0.11 < 0.2 < 1.1$.

In a more explicit way the penalties per constraint with respect to the hierarchy in table 5.5 are shown below (see table 5.6). Note that blank spaces indicate penalty 0 under a (certain) single contraint.

| | *p-Fill*$^C$ ≫ | *p-Parse* ≫ | *p-Fill* $^V$ ≫ | *p-NoCoda* ≫ | *p-Onset* |
|---|---|---|---|---|---|
| c$_1$ | 1 | $(\frac{1}{10})^1 = 0.1$ | | | |
| c$_2$ | | $(\frac{2}{10})^1 = 0.2$ | | | |
| c$_3$ | | $(\frac{1}{10})^1 = 0.1$ | $(\frac{1}{10})^2 = 0.01$ | | |
| c$_4$ | | | | $(\frac{1}{10})^3 = 0.001$ | $(\frac{1}{10})^4 = 0.0001$ |

Table 5.6: Exponential penalties of an OT table

| | *p-Onset* ≫ | *p-NoCoda* ≫ | *p-Fill* $^V$ ≫ | *p-Parse* ≫ | *p-Fill*$^C$ |
|---|---|---|---|---|---|
| c$_1$ | | | | $(\frac{1}{10})^3 = 0.001$ | $(\frac{1}{10})^4 = 0.0001$ |
| c$_2$ | | | | $(\frac{2}{10})^3 = 0.002$ | |
| c$_3$ | | | $(\frac{1}{10})^2 = 0.01$ | $(\frac{1}{10})^3 = 0.001$ | |
| c$_4$ | 1 | $(\frac{1}{10})^1 = 0.1$ | | | |

Table 5.7: Exponential penalties in H'

The penalties are assigned by position. If the hierarchy is changed to *p-Onset* ≫ *p-NoCoda* ≫ *p-Fill*$^V$ ≫ *p-Parse* ≫ *p-Fill*$^C$, we would have the following *system energy* of $\nu$:

Candidate c$_1$: 0.0011<br>
Candidate c$_2$: 0.002<br>
Candidate c$_3$: 0.011<br>
Candidate c$_4$: 1.1

Under this last hierarchy the *preferred* model is c$_1$ (see table 5.7). This model coincides with the *optimal* candidate of the corresponding OT table (see table 2.2, chapter 2).

Instead of *models* we may use *scenarios.* Following formula F2 in definition 4.2 (see chapter 4), we calculate the *optimal* scenario for the PKB conformed with the *p-Con* constraints under the hierarchy *p-Fill*$^C$ ≫ *p-Parse* ≫ *p-Fill*$^V$ ≫ *NoCoda* ≫ *p-Onset.*

For this we need to consider the cost of violating the PKB conforming to the *p-Con* constraints. This cost could be between 1.1111 and 9.9999. We can assume that it is 1.1111 given the fact that in OT the number of violations under a single constraint is not so important in comparison with the hierarchy. The order is kept thanks to the strict domination. If we have two constraints $\mathcal{A}$ and $\mathcal{B}$, and $\mathcal{B}$ is ranked below $\mathcal{A}$, it does not matter if $\mathcal{B}$ has 10, 100 or more violations, it will always be ranked below $\mathcal{A}$ and violations on it will be preferred in order to avoid a violation in $\mathcal{A}$. In a similar way we can talk about $p$-constraints. For that reason it is not important if we say that a certain $p$-constraint has 1 or 2, or more penalties assigned to it.

Recall that the formula F2 introduced in chapter 4 was intended to calculate the cost of a scenario, written $\mathcal{K}_{\Psi}$ ($\Delta$'). The cost of a scenario $\Delta$' in a PKB $\Psi$, i.e. $\mathcal{K}_{\Psi}(\Delta')$, is the sum of the penalties or cost of the formulae of $\psi$ that do not belong to $\Delta$'.

Consider the following scenarios, where $\Delta_{Con} = 1.1111$,

$\Delta_1 = \{$ *Fill*$^V$, *NoCoda*, *Onset* $\}$<br>
$\mathcal{K}_{Con}(\Delta_1) = (\Delta - \Delta_1) = (1.1111 - 0.0111) = \mathbf{1.1}$.

$\Delta_2 = \Delta'_2 + \Delta''_2 = \mathbf{0.2}$. [2]<br>
$\Delta'_2 = \{$ *Fill*$^C$, *Fill*$^V$, *Onset*$\}$<br>
$\mathcal{K}_{Con}(\Delta'_2) = (\Delta - \Delta'_2) = (1.1111 - 1.0111) = 0.1$.<br>
$\Delta''_2 = \{$ *Fill*$^C$, *Fill*$^V$, *Onset* $\}$<br>
$\mathcal{K}_{Con}(\Delta''_2) = (\Delta - \Delta''_2) = (1.1111 - 1.0111) = 0.1$.

$\Delta_3 = \{$ *Fill*$^C$, *NoCoda*, *Onset* $\}$<br>
$\mathcal{K}_{Con}$ $(\Delta_3) = (\Delta - \Delta_3) = (1.1111 - 1.0011) = \mathbf{0.11}$.

$\Delta_4 = \{$ *Fill*$^C$, *Parse*, *Fill*$^V$ $\}$<br>
$\mathcal{K}_{Con}(\Delta_4) = (\Delta - \Delta_4) = (1.1111 - 1.11) = \mathbf{0.0011}$.

The *optimal* scenario is $\Delta_4$ with minimal cost. Observe that the scenarios presented above correspond to the output candidates of table 5.5. The *optimal* candidate c$_4$ is the analogue of the *optimal* scenario $\Delta_4$.

[2] The scenario $\Delta_2$ corresponds to candidate c$_2$ in table 5.5. This candidate has two violations. In order to calculate the penalty of the scenario we need to construct subscenarios. Considering all the constraints in the hierarchy, the number of scenarios needed for a candidate will be determined by the constraint that has more violations. Here *p-Parse* has the maximum number of violations, 2 violations, so, we need two scenarios.

# Chapter 6

# Conclusions

We have presented a translation of a localist symmetric network called $CV_{net}$ into penalty logic. The link *optimal* candidate-*minimal energy* E-*preferred* model has permited us to present an example of how the analogue of an *optimal* candidate arises in the penalty logic environment.

We think that the translation may provide new insights about the encoding of the universal constraints of OT. But it is important to point out that we have assumed through $CV_{net}$ that universality should be implemented by bound weights. This alternative for dealing with universality, even when interesting is not unique. Alternative approaches may be given, for example, in terms of assemblies.

Furthermore, we should observe the localist character of $CV_{net}$ on which our translation is based. Although it may be seen as a straightforward mechanism for capturing the constraints of OT, a distributive representation may also be an interesting option to explore. Perhaps from the point of view of how our brain manages information it might be a more plausible alternative.

What we have presented in the previous pages is just a suggestion of how to re-think the encoding of the universals of OT. But we should stress that the suggestion is not linked *per se* to the nativist account. With the assumption of the same elements for the translation independently of whether they constitute the common structure that all languages share, the translation works. In this manner, the proposal becomes more fruitful.

# Bibliography

[1] Archangeli, Diana(1997). An Introduction to Linguistics in the 1990s. In D. Archangeli and D.T. Langedoen, Terence (eds.), *Optimality Theory: An Overview.* Malden, MA: Blackwell; pp. 1-32.

[2] Bechtel, William and Abrahansem, Adele (2002). *Connectionism and the Mind, Parallel Processing, Dynamics, and Evolution in Networks.* $2^{nd}$ edition. Oxford: Blackwell.

[3] Blutner, Reinhard (2001). Language and Optimality. Seminar at Humboldt University/University of Oslo/ESSLLI, Helsinki.

[4] Blutner, Reinhard (2005). Neural Networks, Penalty Logic and Optimality Theory. Amsterdam: UVA, ILLC Prepublications.

[5] Chomsky, N. (1965). *Aspectos de la Teoría de la Sintaxis* Translation by Carlos Pelegrín Otero from *Aspects of the Theory of Syntax*, Cambridge: MA: The MIT Press. Barcelona: Gedisa, 1999.

[6] Chomsky, N. (1980). *Rules and Representations.* Oxford: Basil Blackwell.

[7] Chomsky, N. (1981a). *Lectures on Government and Binding.* Dordrecht: Foris.

[8] Chomsky, N. (1981b). *Principles and Parameters in Syntactic Theory.* In N. Hornstein and D. Lightfood (eds), *Explanations in Linguistics.* London: Longman.

[9] Clements, G.N. and Keyser, S.J. (1983). *CV Phonology.* Cambridge, MA: The MIT Press. Ch. 1 and 2.

[10] Cook, Vivian and Newson, Mark (1998). *Chomsky Universal Grammar, An Introduction.* Reprinted from $2^{nd}$ edition. Oxford: Blackwell.

[11] Elman, J., Bates, E., Johnson, M., Karmiloff-Smith, Parisi, D. and Plunkett, K. (2001). *Rethinking Innateness, A Connectionist Perspective on Development.* Cambridge, MA: The MIT Press.

[12] Elman, Jeffrey and Plunkett, Kim (1997). *Exercises in Rethinking Innateness, A Handbook for Connectionist Simulations* Cambridge, MA: The MIT Press.

[13] Jakobson, Roman (1962). Typological Studies and their Contribution to Historical Comparative Linguistics. In *Selected Writings 1: Phonological Studies*. The Hague: Mouton.

[14] Féry, Caroline and Van de Vijver, Ruben (2003). Overview. In C. Féry and R. Van de Vijver (eds), *The Syllable in Optimality Theory*. Cambridge: Cambridge University Press; pp. 3-19.

[15] Kager, René (1999a). Conflicts in Grammars. In *Optimality Theory*, Cambridge: Cambridge University Press; pp. 1-51.

[16] Kager, René (1999b). The Typology of Structural Changes. In *Optimality Theory*, Cambridge: Cambridge University Press; pp. 52-90.

[17] McCarthy, John (2002). *A Thematic Guide to Optimality Theory*. Cambridge: Cambridge University Press.

[18] McCarthy, John, and Prince, Alan (1995). *Faithfulness and Reduplicative Identity*. In J. Beckman, L. Walsh Dickey and S. Urbanczyk, *University of Massachusetts Ocasional Papers in Linguistics 18: Papers in Optimality Theory*, pp. 249-384. Amherst, MA: GLSA, University of Massachusetts. Rutgers Optimality Archive 60.

[19] Pinkas, Gadi (1995). Reasoning, Nonmonotonicity and Learning in Connectionist Networks that Capture Propositional Knowledge. *Artificial Intelligence* 77, pp. 203-247.

[20] Prince, Alan, and Smolensky, Paul (1993/2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, Ma: Blackwell. Technical Report, Linguistic Department, Rutgers University, New Brunswick, NJ and Deparment of Computer Science, University of Colorado at Boulder, 1993. Rutgerst Optimality Archive 537 (2002).

[21] Prince, Alan, and Smolensky, Paul (1997). Optimality: From Neural Networks to Universal Grammar. In *Science*, New Series, vol. 275, no. 5306, March 14; pp. 1604-1610.

[22] Prince, Alan (2002). Anything goes. Rutgers Optimality Archive ROA-536.

[23] Prince, Alan, and Smolensky, Paul. To appear. Optimality Theory in Phonology. In William J., Frawledy (ed), *International Encyclopedia of Linguistics*. Oxford: Oxford University Press.

[24] Rumelhart and McClelland (1986). Nativism vs. Empiricism. In Rumelhart, McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: *Foundations*; pp. 139-142. Cambridge, MA: The MIT Press.

[25] Smolensky, Paul and Legendre, Géraldine (2005). To appear. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, vol. 1: *Cognitive Architecture.* Cambridge, MA: The MIT Press.