Lecture 1a: OT – An Introduction

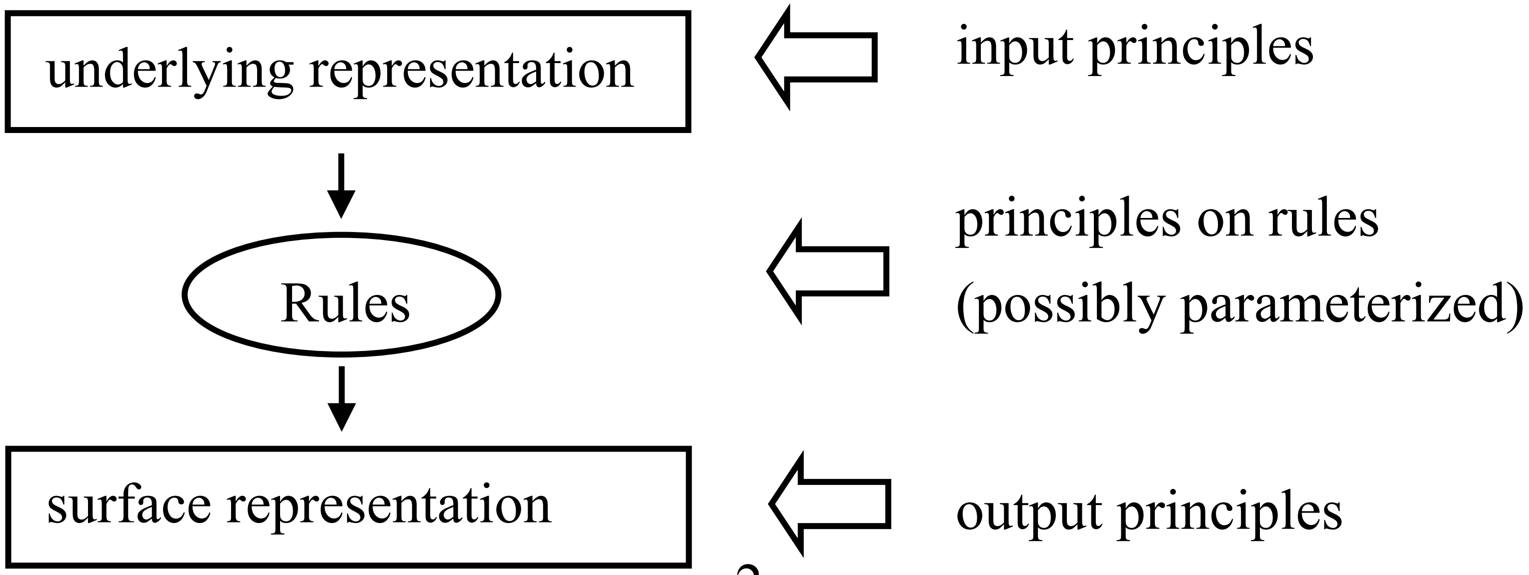
1. Generative linguistics and OT
2. Ethics for robots: a first illustration of OT
3. Voicing contrasts in Dutch and English
4. Basic architecture of standard OT
5. Historical antecedents of OT
6. The rise of OT
7. OT – a new paradigm in linguistics?

1 Generative Linguistics and OT

In Generative Linguistics all the constraints have been viewed *inviolable* within the relevant domain (phonology, syntax).

Basic Picture
(1960s)

Inflation of Principles
(1970s and 1980s)



underlying representation

Rules

surface representation

input principles

principles on rules
(possibly parameterized)

output principles

Standard Scenario of grammatical explanation

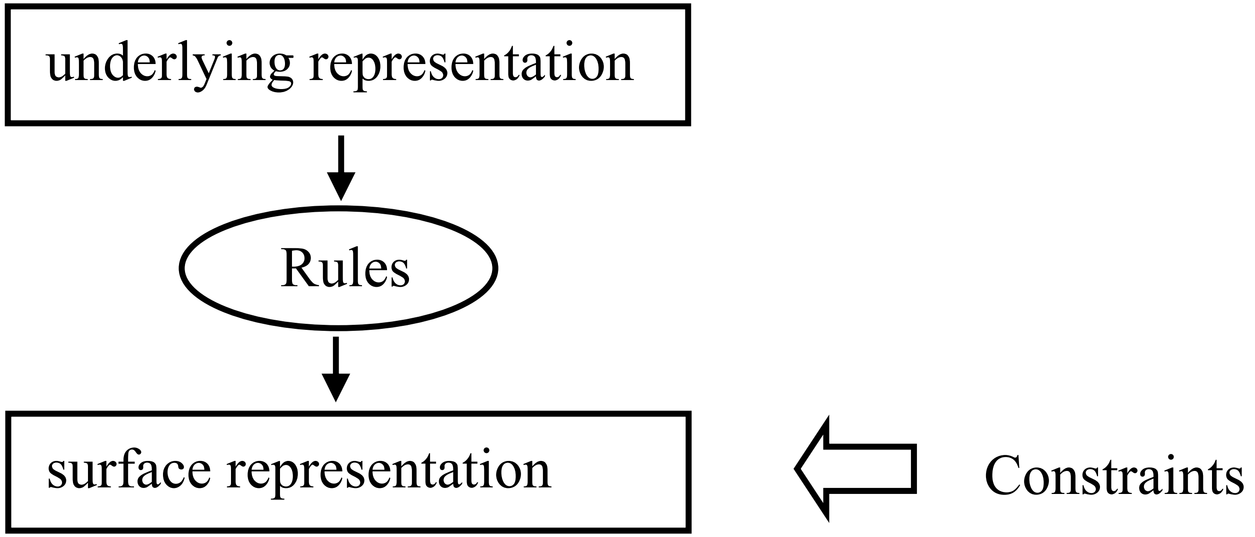
Separation: The status of a particular form with respect to a particular constraint does not depend on the status of any other form with respect to that constraint. In this sense, forms are *separated* from each other.

Inviolability: Constraints, rule systems and principles are *inviolable*. If a form violates a particular principle that violation has an effect on the grammatical status of the object.

Prince & Smolensky: ‘Optimality Theory’

(Arizona Phonology Conference in Tucson, April 1991).

Surface forms of language reflect resolution of conflicts between competing (violable) constraints



underlying representation

Rules

surface representation

Constraints

Optimality Scenario of grammatical explanation

Connection: The status of a particular form with respect to a particular constraint is determined by comparing it with the analysis of other objects. The grammar favours the competitor that best satisfies the constraint. In this sense, forms are *connected* with each other.

Violability: Constraints, rule systems and principles are *violable*. If a form violates a particular constraint C, but no competing form present a lesser violation, that violation of C may result in no detectable deviance.

2 Ethics for robots: A first illustration of OT

Isaac Asimov described what became the most famous view of ethical rules for robot behaviour in his “three laws of robotics” (Thanks to Bart Geurts for drawing my attention to this example):

> ***Three Laws of Robotics:***
>
> *1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.*
>
> *2. A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.*
>
> *3. A robot must protect its own existence, as long as such protection does not conflict with the First or Second Law.*
>
> (Asimov, Isaac: *I, Robot*. Gnome Press 1950)

Analysis

This sentence actually contains three independent constraints:

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings.
3. A robot must protect its own existence.

From an optimality theory point of view, we can think of this as three constraints, where each one overrides the subsequent. The effect of overriding is described by a ranking of the constraints:

$$1 \gg 2 \gg 3,$$

i.e.: ***INJURE HUMAN** ≫ **OBEY ORDER** ≫ **PROTECT EXISTENCE**

Story A: ***H**uman* says to ***R**obot*: Kill my wife!

1. ***R*** kills ***H***'s wife
2. ***R*** kills ***H*** (who gave him the order)
3. ***R*** doesn't kill anyone
4. ***R*** kills himself.

Standard optimality tableau
(☞ marks the optimal candidate, "***!**" the fatal constraint violation):

TABLEAU FOR STORY A	***INJURE HUMAN**	**OBEY ORDER**	**PROTECT EXISTENCE**
1. ***R*** kills ***H***'s wife	*!		
2. ***R*** kills ***H***	*!	*	
☞ 3. ***R*** doesn't kill anyone		*	
4. ***R*** kills himself		*	*!

Comment

In the example, the story relates to a certain situation type that generates the possible reactions 1-4.

R'*s** optimal reaction to ***H's order is to do nothing (line 3). All other reactions are *suboptimal*.

The indication of fatal constraint violation isn't part of the tableaus. It is only to shift the reader's attention to the crucial points.

Story B: ***Human*** says to ***Robo***t: Kill my wife or I kill her!

TABLEAU FOR STORY B	***Injure Human**	**Obey Order**	**Protect Existence**
☞ 1. ***R*** kills ***H***’s wife	*		
2. ***R*** kills ***H***	*	*	
3. ***R*** doesn’t kill anyone	*	*	
4. ***R*** kills himself	*	*	*

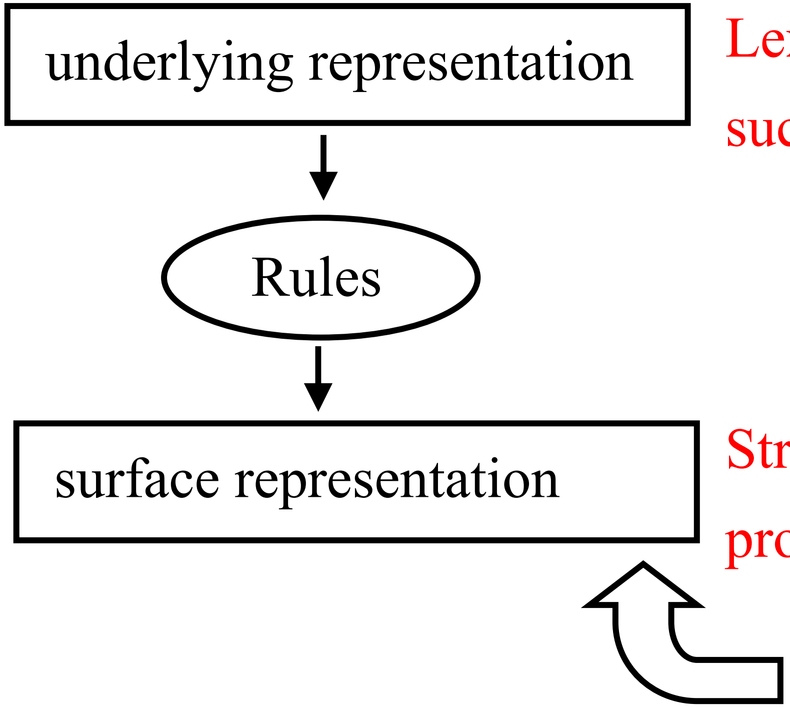
_R_’s optimal reaction to ***H***’s order is to kill ***H***’s wife.

Story C: ***H**uman* says to ***R**obot*: Kill my wife or I destroy you!

TABLEAU FOR Story C	***Injure Human**	**Obey Order**	**Protect Existence**
1. ***R*** kills ***H***'s wife	*		
2. ***R*** kills ***H***	*	*	
☞ 3. ***R*** doesn't kill anyone		*	*
☞ 4. ***R*** kills himself		*	*

There are two optimal reaction to ***H***'s order: ***R*** does nothing (then he is killed by ***H***), or he kills himself.

3 Voicing contrasts in Dutch and English



underlying representation

Lexically motivated strings of segments, such as /bɛd/

Rules

surface representation

Strings of segments related to pronunciation, such as [bɛt]

Constraints

(a. markedness b. mapping)

Phenomenon

Coda obstruents are voiceless in Dutch but voiced in English. Consequently, Dutch neutralizes voicing contrasts in final obstruents and English preserves them:

(1)	a.	/bɛd/	[bɛt]	‘bed’	*Dutch*
	b.	/bɛd-ən/	[bɛ.dən]	‘beds’	
	c.	/bɛt/	[bɛt]	‘(I) dab’	
	d.	/bɛt-ən/	[bɛ.tən]	‘(we) dab’	
(2)	a.	/bɛd/	[bɛd]	‘bed’	*English*
	b.	/bɛt/	[bɛt]	‘bet’	

Two types of constraints

In OT Phonology, we have two kinds of constraints, *markedness conditions*, which evaluate the complexity of the output, and *mapping constraints* which evaluate the difference between input and output.

Markedness Condition

Obstruents must not be voiced in coda position **CODA/*VOICE**

Mapping Constraints

The specification for the feature VOICE of an input segment must be preserved in its output correspondent **FAITH[VOICE]**

Ranking A: CODA/*VOICE ≫ FAITH[VOICE]

Input: /bɛd/	CODA/*VOICE	FAITH[VOICE]
1 ☞ [bɛt]		*
2 [bɛd]	*!	

Input: /bɛt/	CODA/*VOICE	FAITH[VOICE]
1 ☞ [bɛt]		
2 [bɛd]	*	*

This ranking describes the situation in Dutch where voicing contrasts in final obstruents are neutralized.

Ranking B: FAITH[VOICE] ≫ CODA/*VOICE

Input: /bɛd/	FAITH[VOICE]	CODA/*VOICE
1 [bɛt]	*!	
2 ☞ [bɛd]		*

Input: /bɛt/	FAITH[VOICE]	CODA/*VOICE
1 ☞ [bɛt]		
2 [bɛd]	*	*

This ranking describes the situation in English where voicing contrasts in final obstruents are preserved.

What does this example illustrate?

- Two types of constraints in phonology: markedness conditions and mapping constraints (*faithfulness* constraints).
- Markedness is a grammatical factor that exert pressure toward unmarked structure.
- Faithfulness is a grammatical factor that exert pressure toward preserving lexical contrasts.
- Constraints are conflicting. There is no such thing as a ‘perfect’ output. Violations of lower ranked constraints may be tolerated in order to satisfy a higher ranked constraint.

- The Grammar selects an *optimal* output, i.e. an output that best satisfies the system of ranked constraints. More formally:

> A candidate w is considered to be optimal iff for each competitor w’, the constraints that are lost by w must be ranked lower than at least one constraint lost by w’.

- The constraints are universal, their ranking is language particular.

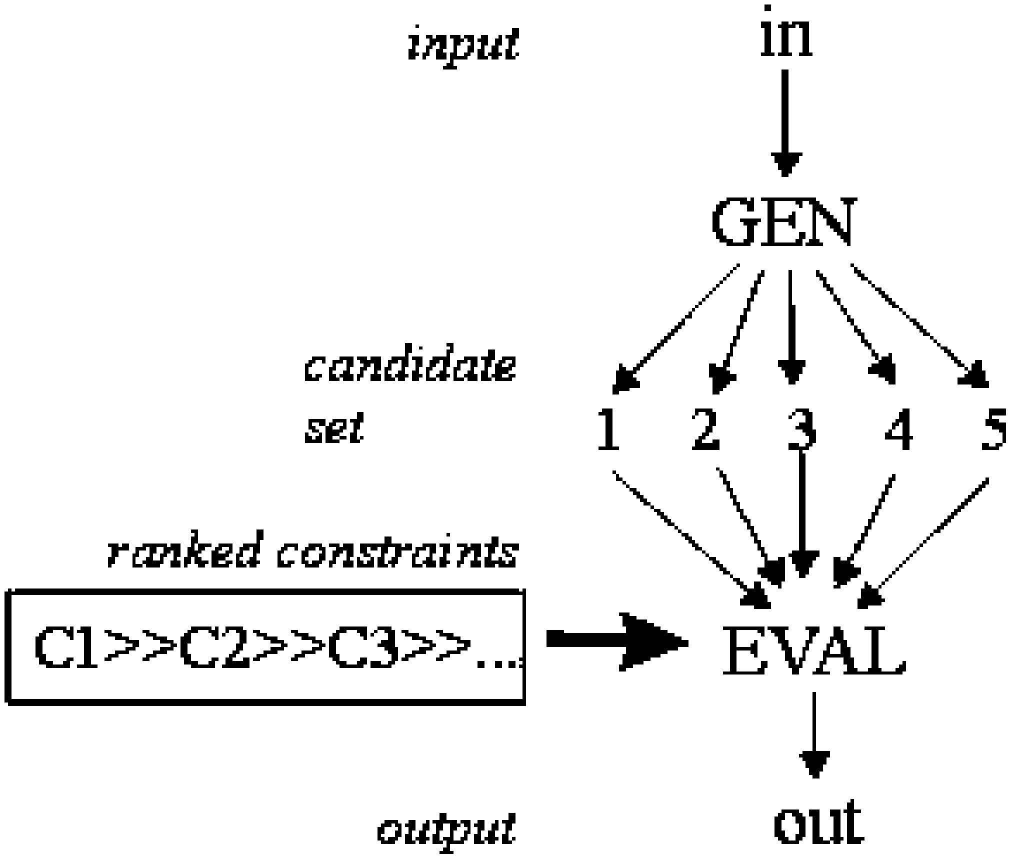
Suggested hypotheses

- Considering all rankings of a given system of (universal) constraints provides a system of language universals for the domain under discussion.
- The different rankings of some (sub)system of constraints provide a typology of natural languages (*factorial typology*).
- Ungrammatical outputs (*) *out* are explained by ‘blocking’: there is an alternate output that satisfies the system of ranked constraints better than *out*.

4 Basic architecture of standard OT

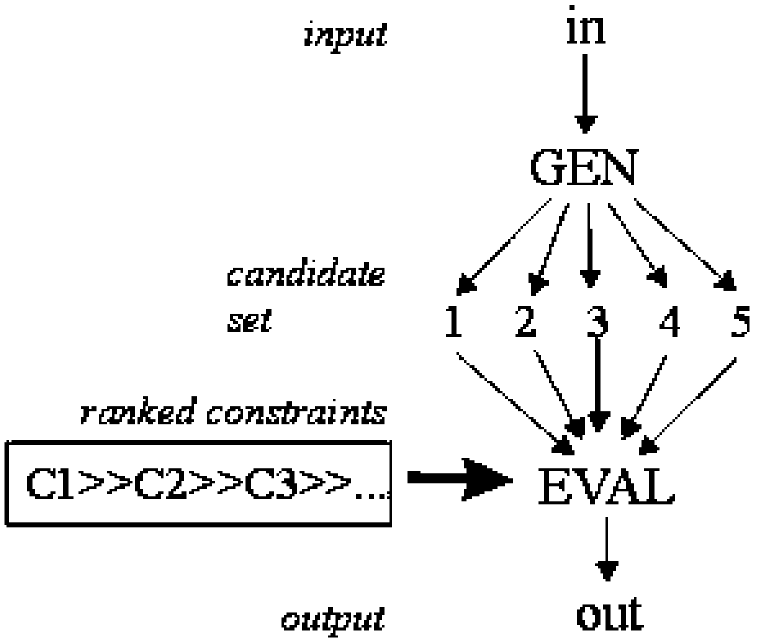
The GENerator
determines the possible inputs, the possible outputs, and the possible correspondences between inputs and outputs. For a given input, GEN creates a candidate set of possible outputs.

OT doesn't provide a 'theory' for GEN, rather it presupposes it. (OT is not a theory of representations!)



The universal CONstraint set is assumed to be part of our innate knowledge of language. Each constraint can be seen as a *markedness statement*. Constraints can be ranked. This reflects the relative importance of the different markedness statements.

EVALuation is a mechanism which selects the optimal candidate(s) from the candidate set generated by GEN. **EVAL** makes use of the ranking of the violable constraints. The optimal output, the one that is selected by **EVAL** is the one that best satisfies these constraints.



5 Historical antecedents of OT

- Panini’s principle in **Phonology**: The application of a rule depends on the failure of a more specific competing rules to apply.

- Specificity in **Morphology**: the most specific vocabulary entry among a set of competitors takes precedence over less specified entries.

- Markedness Theory of Generative **Grammar**

- Hypothetical **Reasoning** (Nicholas Rescher, 1964)

  When Verdi and Bizet were compatriots, then…

  {comp(v,b)↔country(v)=country(b), country(v)=It, country(b)=Fr}

  Definition are ranked higher than facts!

- In **Pragmatics**, the Gricean conversational maxims license an utterance of a particular proposition in a given context only if it fares better (with respect to relevance, for example) than a set of competitors.

- Garden path phenomena in **Natural Sentence Processing**.

  The boat floated down the river sank / and sank

  (based on preferences for the resolution of local ambiguities)

6 The rise of OT

- **The first papers**

  - Alan Prince & Paul Smolensky (1993): Optimality theory: Constraints interaction in generative grammar. *Phonology*

  - John McCarthy & Alan Prince (1993b): Prosodic morphology I: constraint interaction and satisfaction. *Morphology*

- **OT and syntax**

  - Jane Grimshaw 1997: Projection, heads and optimality.

  - Pilar Barbosa, Danny Fox, Paul Hagstrom, Martha McGinnis, & David Pesetsky (eds.): Is the best good enough?

- **Semantics and Interpretation**
  - Helen de Hoop & Henriette de Swart (Eds.) (2000): Papers on Optimality Theoretic Semantics (*J. of Semantics* 17)
  - Reinhard Blutner & Henk Zeevat (Eds.) (2003): Optimality Theory and Pragmatics (Palgrave Macmillan)

Rutgers Optimality Archive

http://roa.rutgers.edu

7 OT - a new paradigm in linguistics?

- Overcoming the gap between competence and performance
- New, powerful learning theory that implicitly makes use of negative examples
- Based on a connectionist architecture (Smolensky's harmony theory). OT aims to integrate symbolist and sub-symbolist (connectionist) systems.
- Interesting from a computational perspective (robust parsing)
- Interesting from an evolutionary perspective (e.g. language change)

Lecture 1b: Phonology of the Syllable

1. Inputs and outputs
2. The optimal correspondence between input and output
3. The Jacobson Typology
4. Conclusions

1 Inputs and outputs

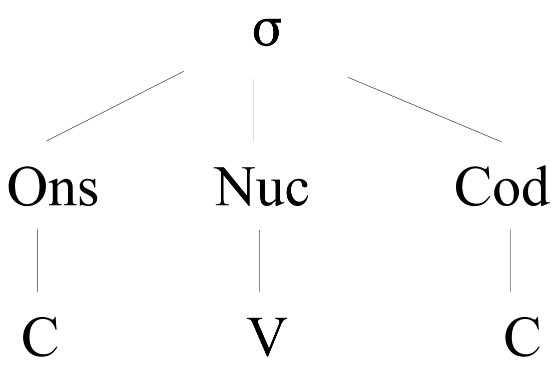
- Inputs are typically taken as simple strings of segments. This strings have to be motivated by morphology.

- Outputs are taken as syllabified strings.

- The output of the phonology is subject to phonetic interpretation. Underparsed segments ⟨x⟩ are not phonetically realized (deletion). Overparsed elements □ are phonetically realized through some process of filling in default featural values (epenthesis)

The structure of the output: Syllables

- Adopt the (simplifying) analysis that the syllable node σ must have a daughter *Nuc* and may have as leftmost and rightmost daughters the nodes *Ons* and *Cod*.
- The nodes *Ons*, *Nuc*, and *Cod*, in turn, may each dominate C´s and V´s, or they may be empty.
- For simplifying further we assume that *Nuc* dominates exactly one V, and *Ons* and *Cod* dominate at most one C.

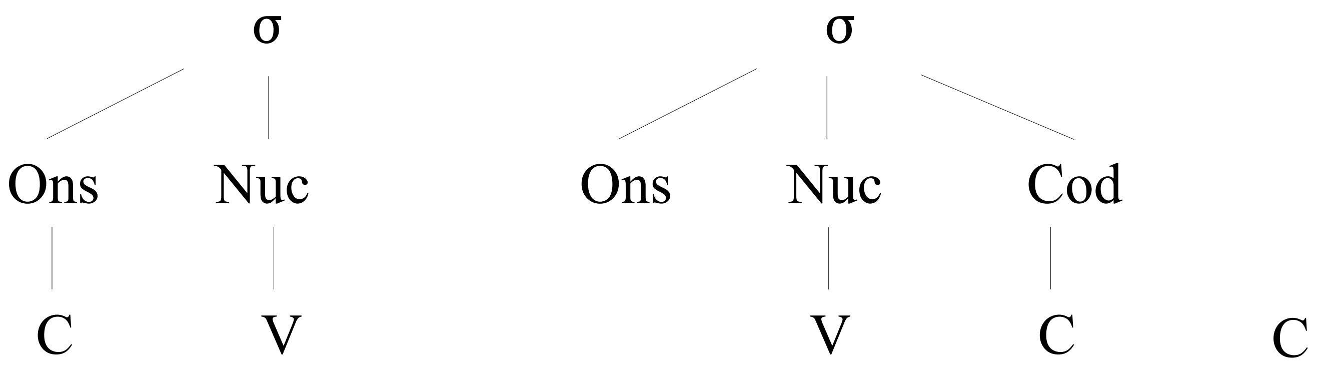


σ

Ons Nuc Cod

C V C

Tree and string notation



.CV.□VC.⟨C⟩

.X. the string X is a syllable

⟨x⟩ the element x has no mother, it is free (not syllabified)

□ a node Ons, Nuc, or Cod is empty

Example

Input	*Output*	*Phonetic*
/no-N-koma-i/	.noɳ.ko.ma.□i. *.noɳ.ko.ma.i.	noɳkoma**t**i *noɳkomai
Consonant epenthesis in Axininca Campa (*noɳkomati* ‘he will paddle’)		

The Generator

It can be assumed to be relatively free. Each possible input is paired with each possible output supposed the corresponding sequences of the terminal elements agree (ignoring □´s).

input: /VCVC/

outputs	a.	.V.CVC.	an onsetless open syllable followed by a closed syllable
	b.	⟨V⟩.CV.⟨C⟩	one open syllable; the initial V and final C are not parsed into syllable structure; this is indicated by ⟨ ⟩
	c.	.□V.CV.⟨C⟩	a sequence of two open syllables. The onset of the first syllable is unfilled (notated □). Phonetically, this is realized as an epenthetic consonant.

2 The optimal correspondence between input and output

Some typical properties of syllables (Markedness Conditions)

Syllables must have onsets ONSET

Syllables must not have a coda NOCODA

Mapping Constraints

No changes in the mapping from input to output FAITHFULNESS

- Underlying segments must be parsed into syllable structure PARSE
- Syllable positions must be filled with underlying segments FILL

3 The Jacobson Typology

> *There are languages lacking syllables with initial vowels and/or syllables with final consonants, but there are no languages devoid of syllables with initial consonants or of syllables with final vowels*. (Jakobson 1962: 526)

These constraints yield exactly four possible systems:

		Onsets	
		required	optional
Codas	forbidden	CV *Senufo (Guinea)*	(C)V Hawaiian
	optional	CV(C) Yawelmani (Cal)	(C)V(C) English

It excludes
V, VC, V(C),
CVC, (C)VC

Explaining the Jacobson typology

Consider the system of constraints {FAITH , ONSET, NOCODA}

A: Ranking FAITH ≫ ONSET, NOCODA

Input: /pipaptaop/	FAITH	ONSET	NOCODA
1 ☞ pi.pap.ta.op		*	**
2 pip.ap.ta.op		**	***
3 pi.pa.⟨p⟩.ta.□o.⟨p⟩	***		
4 pi.pap.ta.□op	*		**
5 pi.pa.⟨p⟩.ta.o⟨p⟩	**	*	

The optimal output realizes syllables (C)V(C)

B: Ranking Onset, NoCoda ≫ Faith

Input: /pipaptaop/			Onset	NoCoda	Faith
1		pi.pap.ta.op	*	**	
2		pip.ap.ta.op	**	***	
3	☞	pi.pa.⟨p⟩.ta.□o.⟨p⟩			***
4		pi.pap.ta.□op		**	*
5		pi.pa.⟨p⟩.ta.o⟨p⟩	*		**

The optimal output realizes syllables CV

C: Ranking ONSET ≫ FAITH ≫ NOCODA

Input: /pipaptaop/			ONSET	FAITH	NOCODA
1		pi.pap.ta.op	*		**
2		pip.ap.ta.op	**		***
3		pi.pa.⟨p⟩.ta.□o.⟨p⟩		***	
4	☞	pi.pap.ta.□op		*	**
5		pi.pa.⟨p⟩.ta.o⟨p⟩	*	**	

The optimal output realizes syllables CV(C)

D: Ranking NoCoda ≫ Faith ≫ Onset

Input: /pipaptaop/			NoCoda	Faith	Onset
1		pi.pap.ta.op	**		*
2		pip.ap.ta.op	***		**
3		pi.pa.⟨p⟩.ta.□o.⟨p⟩		***	
4		pi.pap.ta.□op	**	*	
5	☞	pi.pa.⟨p⟩.ta.o⟨p⟩		**	*

The optimal output realizes syllables (C)V

4 Conclusion

Since ONSET and NOCODA don’t directly interact there are four possible empirically different rankings of the system {FAITH , ONSET, NOCODA} repeated in the table:

Rankings		Types
A	FAITH ≫ ONSET, NOCODA	(C)V(C) *English*
B	ONSET, NOCODA ≫ FAITH	CV *Senufo*
C	ONSET ≫ FAITH ≫ NOCODA	CV(C) *Yawelmani*
D	NOCODA ≫ FAITH ≫ ONSET	(C)V *Hawaiian*

The four possible rankings describe all and only the possible syllable type systems.

	Onset ≫ Faith	Faith ≫ Onset
NoCoda ≫ Faith	CV *Senufo (Guinea)*	(C)V *Hawaiian*
Faith ≫ NoCoda	CV(C) *Yawelmani (Cal)*	(C)V(C) *English*

In general, for any set of freely rankable constraints, OT predicts the possibility of languages corresponding to each possible ranking. This is called the *Factorial Typology*. The factorial typology that corresponds to the Jacobson typology was proposed first by Prince & Smolensky (1993)