A Crash Course in Optimality Theory

Jochen Trommer
jtrommer@uni-leipzig.de

University of Leipzig
Department of Linguistics

Concatenative Approaches to
Nonconcatenative Morphology
EGG 2008

Optimality Theory (in General)

a formalism which allows to choose between alternative options

for a given situation

on the basis of (potentially) contradicting

ranked preferences (constraints)

Optimality Theory (Phonology)

An application of this formalism to Phonology

+

Substantial assumptions on possible representations and constraints

OT in General: Robot Ethics

Isaac Asimovs ethical rules for the behaviour of robots (the “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.

Robot Ethics and Potential Conflicts

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**.

Robot Ethics in OT

Instead of stipulating in single rules

how they interact with other rules

the rules are formulated as completely general constraints

which are ranked among each other

Robot Ethics in OT

***INJURE HUMAN :**	A robot may not injure a human being or, through inaction, allow a human being to come to harm.
OBEY ORDER:	A robot must obey the orders of human beings.
PROTECT EXISTENCE:	A robot must protect its own existence.

Ranking:

*INJURE HUMAN: $\gg$ OBEY ORDER $\gg$ PROTECT EXISTENCE

Story A:

Human 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

Evaluating Possible Outcomes

1. Setting up a tableau
2. Assigning violation marks
3. Eliminating suboptimal candidates

Step 1: Setting up a Tableau for Story A

Input: H says to **R**: Kill my wife!

	*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s wife			
R kills **H**			
R doesn't kill anyone			
R kills himself			

Step 2: Assigning Violation Marks for Story A

Input: H says to **R**: Kill my wife!

	*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s wife	*		
R kills **H**	*	*	
R doesn't kill anyone		*	
R kills himself		*	*

Step 3: Eliminating Suboptimal Candidates for Story A

Input: H says to **R**: Kill my wife!

	*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s wife	*!		
R kills **H**	*!	*	
☞ **R** doesn't kill anyone		*	
R kills himself		*	*!

Optimality Theory: Evaluation of Tableaus

for the constraints x $\in$ 1 . . . n

- remove all candidats from the candidate set, which are suboptimal for constraint **x**

until there is only 1 candidate left

Optimality Theory: Evaluation of Tableaus

- Remove all candidates from the candidate set which are suboptimal for constraint **1**
  If there is only 1 candidate left: Stop!
- Remove all candidates from the candidate set which are suboptimal for constraint **2**
  If there is only 1 candidate left: Stop!
- Remove all candidates from the candidate set which are suboptimal for constraint **3**
  If there is only 1 candidate left: Stop!
- . . .

Step 3: Eliminating Suboptimal Candidates for Story A

Input: H says to **R**: Kill my wife!

		*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
	R kills **H**'s wife	*!		
	R kills **H**	*!	*	
☞	**R** doesn't kill anyone		*	
	R kills himself		*	*!

Optimality Theory: Symbols

*	Star	Constraint Violation
!	Exclamation Mark (after star)	Fatal Constraint Violation
☞	Pointing Hand	Optimal candidate
	Shading	Irrelevant Tableau Cells

Step 3: Eliminating Suboptimal Candidates for Story A

Input: H says to **R**: Kill my wife!

		*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
	R kills **H**'s wife	*!		
	R kills **H**	*!	*	
☞	**R** doesn't kill anyone		*	
	R kills himself		*	*!

Ranking Matters: A Different Ranking for Story A

Input: H says to **R**: Kill my wife!

	OBEY HUMAN	*INJURE ORDER	PROTECT EXISTENCE
☞ **R** kills **H**'s wife		*	
R kills **H**	*!	*	
R doesn't kill anyone	*!		
R kills himself	*!		*

Input Matters: Story B

Input: H says to **R**obot: Kill my wife or I kill her!

	*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
☞ **R** kills **H**'s wife	*		
R kills **H**	*	*!	
R doesn't kill anyone	*	*!	
R kills himself	*	*!	*

Optimality Theory in Phonology

Inputs: Phonological representations from Lexicon + Morphology

Outputs: Modified phonological forms

Final devoicing in German

Rat	[ʀaː**t**]	‘advice’	Rat+es	[ʀaː**t**+əs]	‘advice (gen.)’
Stück	[ʃty**k**]	‘piece’	Stück+es	[ʃty**k**+əs]	‘piece’ (gen.)

Rad	[ʀaː**t**]	‘wheel’	Rades	[ʀaː**d**+əs]	‘wheel’ (gen.)
Tag	[taː**k**]	‘day’	Tages	[taː**g**+əs]	‘day’ (gen.)

Voiced stops get voiceless at the end of a word

Final devoicing in Rule-based Phonology

[+voiced] ➔ [–voiced] /___ #

/ʀa:d/ ⇒ /ʀa:t/

Final Devoicing in Optimality Theory

Input: tu:gend	
☞	a. tu:gent
	b. du:gent
	c. tu:gend
	d. du:gend

The Basic Conflict in Optimality Theory

Markedness Constraints:	Make things better!
Treue-Beschränkungen:	Don't change!

Constraints

Markedness

***[+voiced]#**	Stops at the word end should not be [+voiced]
IDENT [voiced]	Corresponding input and output sounds should have identical values for the feature [voiced]

Faithfulness

The Interpretation of OT-Constraints

In the literature, OT-Constraints are often formulated as positive requirements,

but **technically** every constraint is a function which assigns a specific number of constraint violations for a specific input-output pair

*[+voiced]#

Input: tu:gend	*[+voiced]#
☞ a. tu:gent	
☞ b. du:gent	
c. tu:gend	*!
d. du:gend	*!

***[+voiced]#**	Stops at the word end should not be [+voiced]
	Count 1 constraint violation for every voiced stop at the end of a word

IDENT [voiced]

	Input: tu:gend	IDENT [voiced]
	a. tu:gent	*!
	b. du:gent	*!*
☞	c. tu:gend	
	d. du:gend	*!

IDENT [voiced]	Corresponding input and output sounds should have identical values for the feature [voiced]
	Count 1 constraint violation for every output sound which has a different value of [voiced] with respect to the corresponding input sound

Final Devoicing in Optimality Theory: Ranking

Input: tu:gend	*[+voiced]#	IDENT [voiced]
☞ a. tu:gent		*
b. du:gent		**!
c. tu:gend	*!	
d. du:gend	*!	*

Final Devoicing in Optimality Theory: Ranking

Input: tu:gend	*[+voiced]#	IDENT [voiced]
☞ a. tu:gent		*
b. du:gent		**!
c. tu:gend	*!	
d. du:gend	*!	*

Optimality-theoretic Constraints . . .

- are universal (innate?)
- phonetically grounded
- freely rankable

Different rankings result
in the grammars of different languages

Final devoicing . . .

- is attested in many unrelated languages
- doesn't have to be learned (Stampe, 1973)
- occurs also in the acquisition of languages without final devoicing (Smith, 1973)

Ranking for Languages without Final devoicing

Input: tu:gend	IDENT [voiced]	*[+voiced]#
a. tu:gent	*!	
b. du:gent	*!*	
☞ c. tu:gend		*
d. du:gend	*!	*