

# Einleitung

Jochen Trommer  
jtrommer@uni-leipzig.de

Universität Leipzig

Optimalitätstheorie

# Ethics for Robots: An illustration of OT

Isaac Asimov described what became the most famous view of ethical rules for robot behaviour in his “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.

# Ethics for Robots: An illustration of OT

Isaac Asimov described what became the most famous view of ethical rules for robot behaviour in his “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.**

# Ethics for Robots 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.

## Ranked:


\*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 Second Law.

# OT-Tableau for Story A

	*Injure HUMAN	OBEY ORDER	PROTECT EXISTENCE
<b>R kills H's wife</b>	*!		
<b>R kills H</b>	*!	*	
 <b>R doesn't kill anyone</b>		*	
<b>R kills himself</b>		*	*!

# Algorithm **A** for finding the optimal candidate

1. **If** there is only one candidate or no constraints:  
all candidates are optimal

2. **Else:**


eliminate all candidates  
which are suboptimal for the highest-ranked constraint

eliminate the highest ranked constraint

Apply **A**

# Story B:



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

	*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
 <b>R kills H's wife</b>	*		
<b>R kills H</b>	*	*!	
<b>R doesn't kill anyone</b>	*	*!	
<b>R kills himself</b>	*	*!	*



# Story C:

Human says to **R**obot: Kill my wife or I destroy you!

	*INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
<b>R</b> kills <b>H</b> 's wife	*!		
<b>R</b> kills <b>H</b>	*!	*	
 <b>R</b> doesn't kill anyone		*	*
 <b>R</b> kills himself		*	*

# Optimality Theory (Prince & Smolensky, 1993)

- Universal Grammar contains a fixed set of constraints
- Constraints are violable, but violation is minimal
- Constraint conflict is resolved by constraint ranking
- Grammars of single languages result from different constraint rankings

# Jakobson's Syllable Typology

- There are languages where syllable onsets are obligatory, but no languages where onsets are impossible
- There are languages where syllable codas are impossible, but no languages where codas are obligatory


# Same as OT Constraints


- **ONSET** (Syllables should have Onsets)
- **NoCODA** (Syllable codas should be avoided)

## + 2 more natural constraints


- **MAX** (Don't delete input segments)
- **DEP** (Don't insert new segments)


# Language without codas

<b>Input:</b> $b_1 a_2 b_3$	NoCODA	DEP	MAX
 $b_1 a_2$			*
$b_1 a_2 b_3 a_4$		*!	
$b_1 a_2 b_3$	*!		


<b>Input:</b> $b_1 a_2$	NoCODA	DEP	MAX
$b_1$			*!
 $b_1 a_2$			
$b_1 a_2 b_3$	*!	*	


# Language with codas

<b>Input:</b> $b_1 a_2 b_3$	MAX	DEP	NoCODA
$b_1 a_2$	*!		
 $b_1 a_2 b_3$			*
$b_1 a_2 b_3 a_4$		*!	


<b>Input:</b> $b_1 a_2$	MAX	DEP	NoCODA
$b_1$	*!		
 $b_1 a_2$			
$b_1 a_2 b_3$		*!	*


# Language with obligatory onset

<b>Input:</b> $a_1b_2a_3$	ONSET	DEP	MAX
 $b_2a_3$			*
$a_1b_2a_3$	*!		
$b_4a_1b_2a_3$		*!	

<b>Input:</b> $b_1a_2$	ONSET	DEP	MAX
$a_2$	*!		*
 $b_1a_2$			
$b_1a_2b_3$		*!	

# Language with optional onset

<b>Input:</b> $a_1 b_2 a_3$	DEP	MAX	ONSET
$b_2 a_3$		*!	
 $a_1 b_2 a_3$			*
$b_4 a_1 b_2 a_3$	*!		

<b>Input:</b> $b_1 a_2$	DEP	MAX	ONSET
$a_2$		*!	*
 $b_1 a_2$			
$b_1 a_2 b_3$	*!		