Agent-based modeling:

Agents with a complex cognitive architecture
(A tutorial)

Tamás Biró

Eötvös Loránd University (ELTE)

Network Theory and Computer Modeling in the Study of Religion
September 1, 2016

Sentential negation (Jespersen's cycle)

	pre-verbal	discontinuous	post-verbal
French	Jeo ne dis	Je ne dis pas	Je dis pas
English	Ic ne secge	Ic ne seye not	I say not
	1. *SN V*	2. *SN V SN*	3. *V SN*

To explain:

- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).

Sentential negation (Jespersen's cycle)

	pre-verbal	discontinuous	post-verbal
French	Jeo ne dis	Je ne dis pas	Je dis pas
English	Ic ne secge	Ic ne seye not	I say not
	1. *SN V*	2. *SN V SN*	3. *V SN*

To explain:

- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).

Sentential negation (Jespersen's cycle)

	pre-verbal	discontinuous	post-verbal
French	Jeo ne dis	Je ne dis pas	Je dis pas
English	Ic ne secge	Ic ne seye not	I say not
	1. *SN V*	2. *SN V SN*	3. *V SN*

To explain:

- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).

Sentential negation (Jespersen's cycle)

	pre-verbal	discontinuous	post-verbal
French	Jeo ne dis	Je ne dis pas	Je dis pas
English	Ic ne secge	Ic ne seye not	I say not
	1. *SN V*	2. *SN V SN*	3. *V SN*

To explain:

- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).

Sentential negation (Jespersen's cycle)

Research questions:

- Why does this language change happen?
- What drives change?
  "Performance errors" as a driving force behind language change?

Methodology:

- Multi-agent simulations

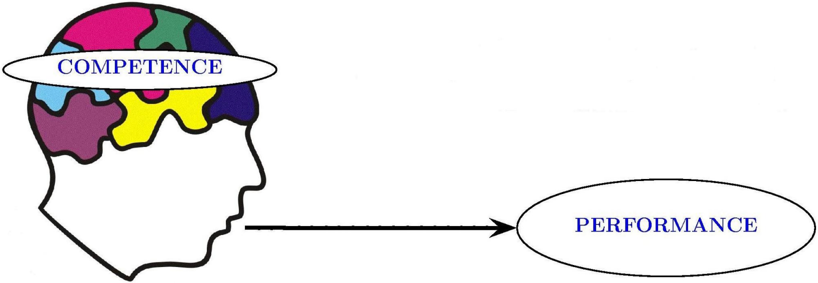
Reference

- A. Lopopolo and T. Biró. 'Language Evolution and SA-OT: The case of sentential negation'. *Computational Linguistics in the Netherlands Journal* 1(2011):21–40.

Learning (what CSR misses from CogSci)

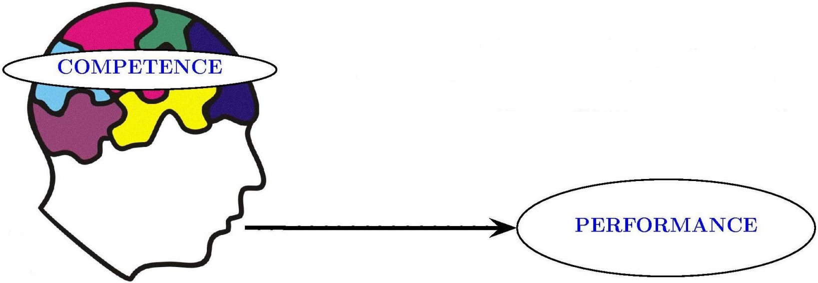
- **Learning:** the algorithm behind / modeling acquisition. A central topic in linguistics and cognitive science, but missing in CSR. See also *machine learning*.

Errors of the mental computation



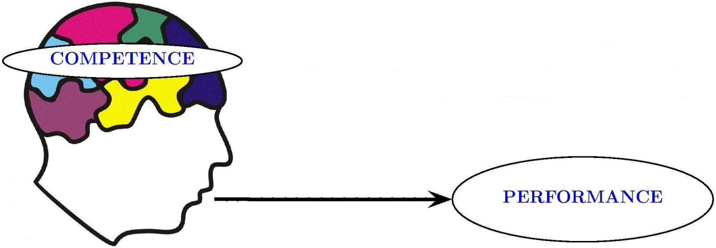
static knowledge processes in the brain

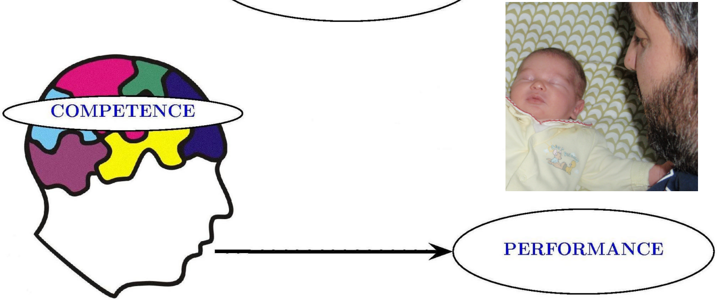
Errors of the mental computation



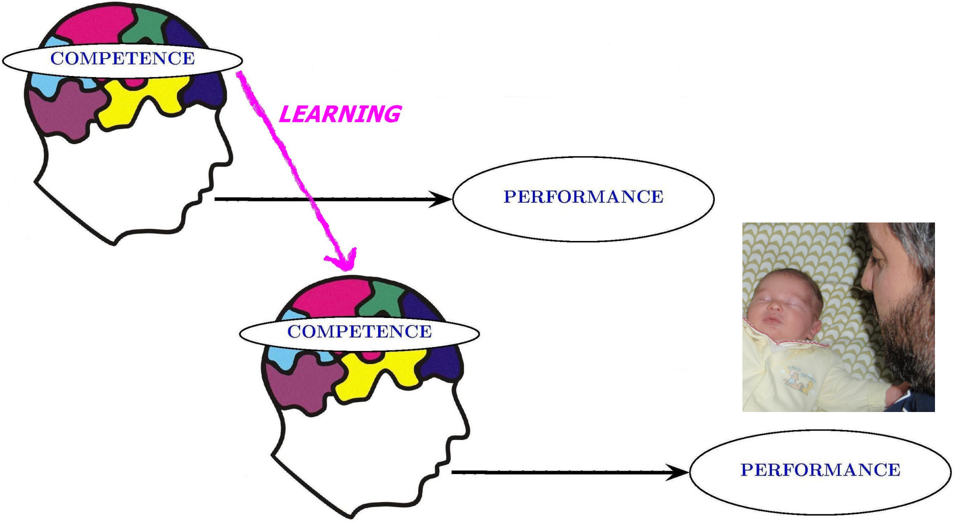
static knowledge processes in the brain

The language acquisition problem

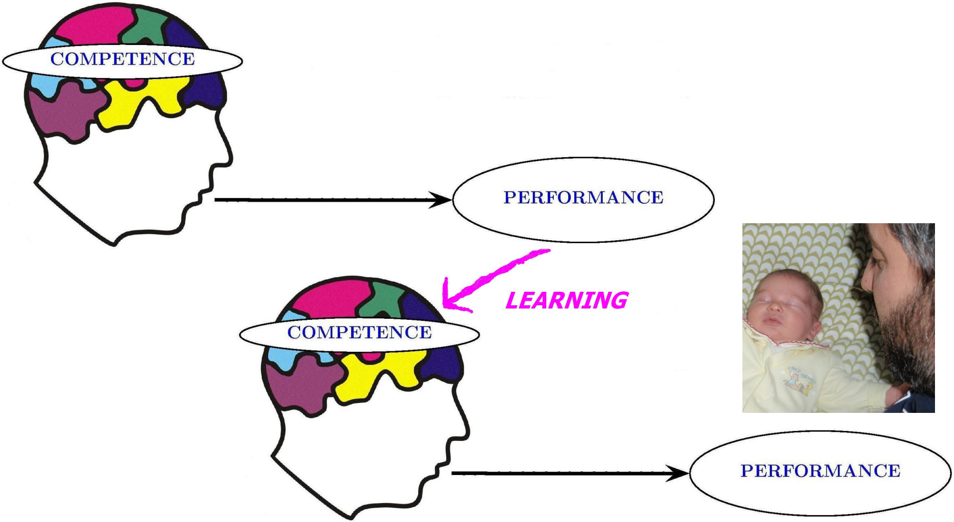




Learning from competence?



Learning from performance!



The *Iterative Learning Model*

- **Learning:** the algorithm behind / modeling acquisition.
  A central topic in linguistics and cognitive science,
  but missing in CSR. See also *machine learning*.
- **Iterative learning:**

  Gen-0 ⟶ *Gen-1* ⟶ *Gen-2* ⟶ *Gen-3* ⟶

- Simon Kirby at al.: language evolution (in biological evolution's timescale). "Learning bottleneck" creates linguistic structure.
- Others: language change (in historical timescale).
  Assumption: language change takes place from generation to generation, due to imperfect acquisition. (Only partly true.)

(Possible) components of the model

Who learns from whom?

- N agents in **one generation.**
- **Series of generations:** language produced by agents in Generation k used as learning data by agents in Gen. $k+1$.

Generation k: a1 a2 a3 a4

Generation $k+1$: b1 b2 b3 b4

- Note the strict intergenerational structure: no learning from grandparents, elder siblings or peers.
- Social structure? More learning data from parents? Learning data with more weight from people with prestige?

(Possible) components of the model

Who learns from whom?

- *N* agents in **one generation.**
- **Series of generations:** language produced by agents in Generation k used as learning data by agents in Gen. $k+1$.

Generation k: a1 a2 a3 a4

Generation $k+1$: b1 b2 b3 b4

- Note the strict intergenerational structure: no learning from grandparents, elder siblings or peers.
- Social structure? More learning data from parents? Learning data with more weight from people with prestige?

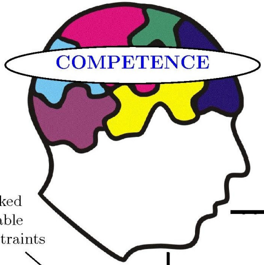
(Possible) components of the model

Who learns at all?

An agent composed of:

- **Knowledge**: a.k.a. competence, grammar, etc.
  Here: Optimality Theory (Prince and Smolensky 1993/2006)
- **Production**: a.k.a. performance, etc.
  Here: Simulated Annealing for Optimality Theory (Biró 2006)
- **Learning**: a.k.a. acquisition, etc.
  Here: online learning algorithms for Optimality Theory (Boersma and Hayes 2001; Magri 2012)

Errors of the mental computation



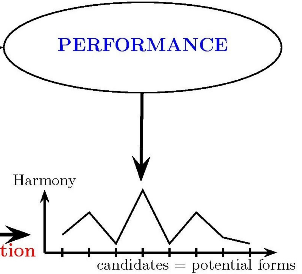
A grammar is a Harmony function on the candidate set, defined by the ranked constraints.
Global optimum: more harmonic than all other candidates.
Local optimum: more harmonic than its neighbours.

Ranked violable constraints

Candidates (potential forms)

/aat/	NoCoda	Parse	Onset
☞ [a.a<t>]		*	**
[a.at]	*!		**
[<a>at]	*!	*	*

implementation



Optimality Theory
grammar
competence model
grammatical form = ☞ (globally) optimal candidate

SA-OT
implementation
performance model
produced forms = globally or locally optimal candidates

Sentential negation: Jespersen's cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

Generation 3 learning from performance pattern of Generation 2. Etc.

Sentential negation: Jespersen's cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

Generation 3 learning from performance pattern of Generation 2. Etc.

Sentential negation: Jespersen's cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: *NEGATION $\gg$ NEGATIONFIRST $\gg$ NEGATIONLAST
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST $\gg$ *NEGATION $\gg$ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

Generation 3 learning from performance pattern of Generation 2. Etc.

Sentential negation: Jespersen's cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

Questions:

1. Has the learning been successful?
   1. grammaticality judgement; 2. grammar; 3. performance pattern.
2. A way to model diachronic change?
3. Learner hears "SN V SN": is it [[SN V] SN] or [SN [V SN]]?

Sentential negation: Jespersen's cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: *NEGATION $\gg$ NEGATIONFIRST $\gg$ NEGATIONLAST
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST $\gg$ *NEGATION $\gg$ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

Questions:

1. Has the learning been successful?
   1. grammaticality judgement; 2. grammar; 3. performance pattern.
2. A way to model diachronic change?
3. Learner hears "SN V SN": is it [[SN V] SN] or [SN [V SN]]?

Sentential negation: Jespersen's cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

Questions:

1. Has the learning been successful?
   1. grammaticality judgement; 2. grammar; 3. performance pattern.
2. A way to model diachronic change?
3. Learner hears "SN V SN": is it [[SN V] SN] or [SN [V SN]]?

Modelling linguistic competence

FAITH[NEG] $\gg$ *NEGATION $\gg$ NEGATIONFIRST $\gg$ NEGATIONLAST

	/pol = neg/	Faith[Neg]	*Neg	NegFirst	NegLast
	[V]	*		*	*
☞	[SN V]		*		*
	[V SN]		*	*	
	[SN V SN]		**		
	[V SN SN]		**	*	
	[SN SN V]		**		*
	[SN V SN SN]		***		
	...				

Lopopolo and Biró (2011), based on Henriëtte de Swart (2010).

Modelling linguistic competence

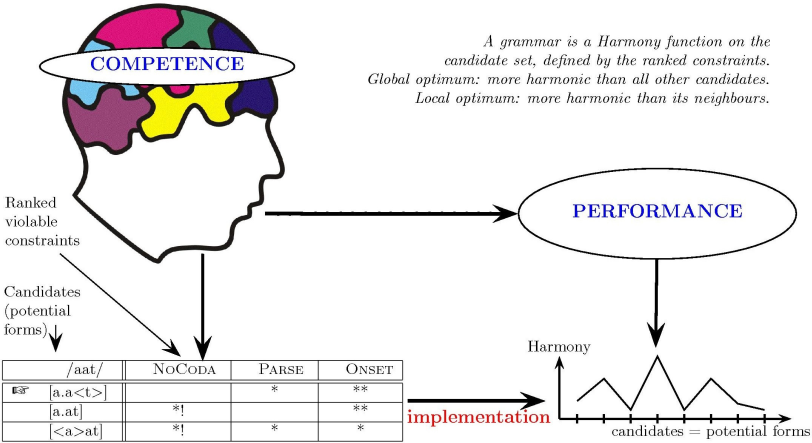
FAITH[NEG] ≫ NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST

	/pol = neg/	Faith[Neg]	NegFirst	*Neg	NegLast
	[V]	*	*		*
☞	[SN V]			*	*
	[V SN]		*	*	
	[SN V SN]			**	
	[V SN SN]		*	**	
	[SN SN V]			**	*
	[SN V SN SN]			***	
	...				

Lopopolo and Biró (2011), based on Henriëtte de Swart (2010).

Errors of the mental computation

A grammar is a Harmony function on the candidate set, defined by the ranked constraints.
Global optimum: more harmonic than all other candidates.
Local optimum: more harmonic than its neighbours.



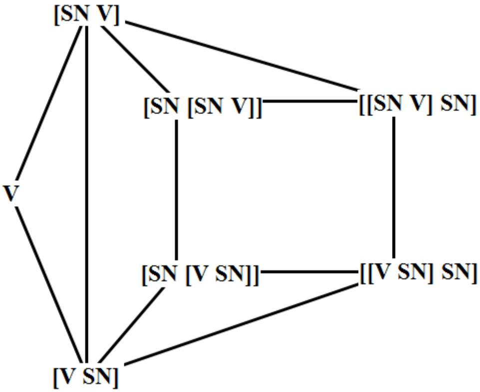
/aat/	NoCoda	Parse	Onset
☞ [a.a<t>]		*	**
[a.at]	*!		**
[<a>at]	*!	*	*

Optimality Theory
grammar
competence model
grammatical form = ☞ (globally) optimal candidate

SA-OT
implementation
performance model
produced forms = globally or locally optimal candidates

Modelling linguistic performance

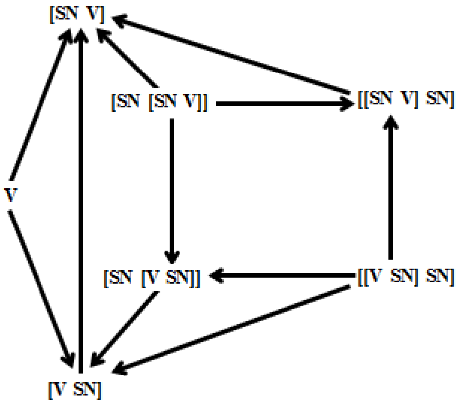
A topology (neighborhood structure) on the candidate set:



Locally optimal forms: are predicted to be the produced forms.

Modelling linguistic performance

FAITH[NEG] ≫ *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST

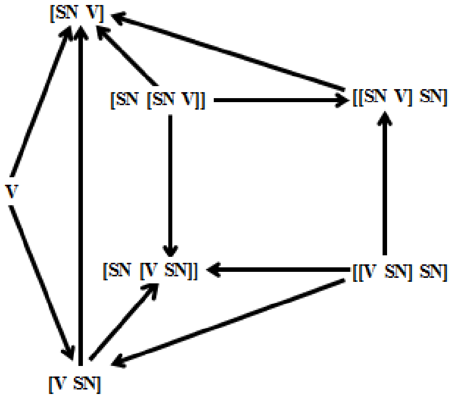


Hierarchy 1: *Neg >> NegFirst >> NegLast

Locally optimal forms: ☞ [SN V].

Modelling linguistic performance

FAITH[NEG] $\gg$ NEGATIONFIRST $\gg$ *NEGATION $\gg$ NEGATIONLAST



Hierarchy 2: NegFirst >> *Neg >> NegLast

Locally optimal forms: ☞ [SN V] and $\sim$ [SN [V SN]].

	Hierarchy	competence	performance
1.	*Neg $\gg$ NegFirst $\gg$ NegLast	pre-verbal	pre-verbal
2.	NegFirst $\gg$ *Neg $\gg$ NegLast	pre-verbal	pre-V and discont.
3.	NegFirst $\gg$ NegLast $\gg$ *Neg	discontinuous	discontinuous
4.	NegLast $\gg$ NegFirst $\gg$ *Neg	discontinuous	discontinuous
5.	NegLast $\gg$ *Neg $\gg$ NegFirst	post-verbal	discont. and post-V
6.	*Neg $\gg$ NegLast $\gg$ NegFirst	post-verbal	post-verbal

Observerd typology: 3 pure types and 2 mixed types.
Predicted typology:

- Traditional OT (H. de Swart): 3 pure types.
- Stochastic OT (H. de Swart): 3 pure types and 3 mixed types.
- SA-OT (Lopopolo and Biró): 3 pure types and 2 mixed types.

	Hierarchy	competence	performance
1.	*Neg $\gg$ NegFirst $\gg$ NegLast	pre-verbal	pre-verbal
2.	NegFirst $\gg$ *Neg $\gg$ NegLast	pre-verbal	pre-V and discont.
3.	NegFirst $\gg$ NegLast $\gg$ *Neg	discontinuous	discontinuous
4.	NegLast $\gg$ NegFirst $\gg$ *Neg	discontinuous	discontinuous
5.	NegLast $\gg$ *Neg $\gg$ NegFirst	post-verbal	discont. and post-V
6.	*Neg $\gg$ NegLast $\gg$ NegFirst	post-verbal	post-verbal

Observerd typology: 3 pure types and 2 mixed types.
Predicted typology:

- Traditional OT (H. de Swart): 3 pure types.
- Stochastic OT (H. de Swart): 3 pure types and 3 mixed types.
- SA-OT (Lopopolo and Biró): 3 pure types and 2 mixed types.

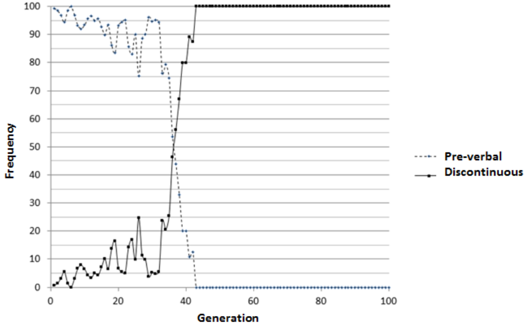
	Hierarchy	competence	performance
1.	*Neg $\gg$ NegFirst $\gg$ NegLast	pre-verbal	pre-verbal
2.	NegFirst $\gg$ *Neg $\gg$ NegLast	pre-verbal	pre-V and discont.
3.	NegFirst $\gg$ NegLast $\gg$ *Neg	discontinuous	discontinuous
4.	NegLast $\gg$ NegFirst $\gg$ *Neg	discontinuous	discontinuous
5.	NegLast $\gg$ *Neg $\gg$ NegFirst	post-verbal	discont. and post-V
6.	*Neg $\gg$ NegLast $\gg$ NegFirst	post-verbal	post-verbal

Observerd typology: 3 pure types and 2 mixed types.
Predicted typology:

- Traditional OT (H. de Swart): 3 pure types.
- Stochastic OT (H. de Swart): 3 pure types and 3 mixed types.
- SA-OT (Lopopolo and Biró): 3 pure types and 2 mixed types.

Iterated learning: reproducing language change (?)

Five agents in each generation. Generations 0 to 100.
Each agent learns from every agent in the previous generation.
Negation types in the “simulated historical corpus”:



A. Lopopolo and T. Biró. ‘Language Evolution and SA-OT: The case of sentential negation’.

Computational Linguistics in the Netherlands Journal 1(2011):21–40.

Conclusions

What is the question / interest:

- **Proceed from the phenomenon:** explaining Jespersen's cycle.
- **Proceed from theory:** role of *errors* $=$ results of imperfect mental computation in language change.
- **Proceed from framework:** the behavior of a certain theoretical, computational, mathematical framework.

Conclusions

Model:

- Agents $\rightarrow$ (un)structured population $\rightarrow$ generation.
- Agents $\rightarrow$ knowledge (competence), production (performance) and learning (acquisition).
- Iterative learning model

Practicalities

Practicalities:

- Developed in own software *OTKit* (`http://www.birot.hu/OTKit/`).
- The more complex a model: the more parameters.
- The convincing force of a complex, still abstract and oversimplified computational model?

Thank you for your attention!

Tamás Biró:

`tamas.biro@btk.elte.hu`

OT KIT

Tools for Optimality Theory

`http://www.birot.hu/OTKit/`

Work supported by:

MARIE CURIE ACTIONS

Amsterdam Center for Language and Communication ACLC

NWO Netherlands Organisation for Scientific Research