Subregular toolkit implemented in Python

Alëna Aksënova

Linguistics department @ SBU

IACS 5-year Review Board

IACS @ SBU

September 8, 2017

A mysterious puzzle

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

A mysterious puzzle

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

A mysterious puzzle

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

Ilocano

2a. bigat
morning

2b. ka-bigat-an
the next morning

A mysterious puzzle

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

Ilocano

2a. bigat
morning

2b. ka-bigat-an
the next morning

2c. *ka^n-bigat-an^n
*the morning after the next one

A mysterious puzzle

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

Ilocano

2a. bigat
morning

2b. ka-bigat-an
the next morning

2c. *ka^n-bigat-an^n
*the morning after the next one

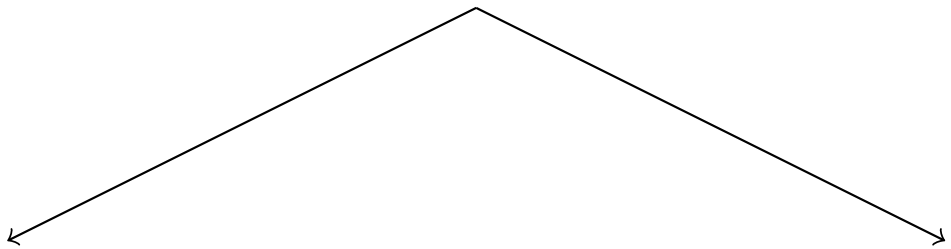
Why is (2c) impossible?

Human language and its complexity

Why does the complexity of human language matter?

Human language and its complexity

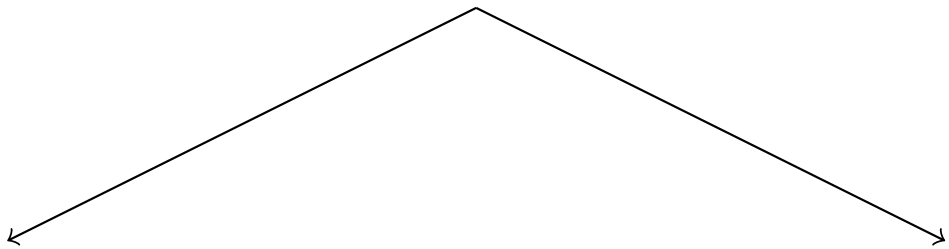
Why does the complexity of human language matter?



Science **Industry**

Human language and its complexity

Why does the complexity of human language matter?



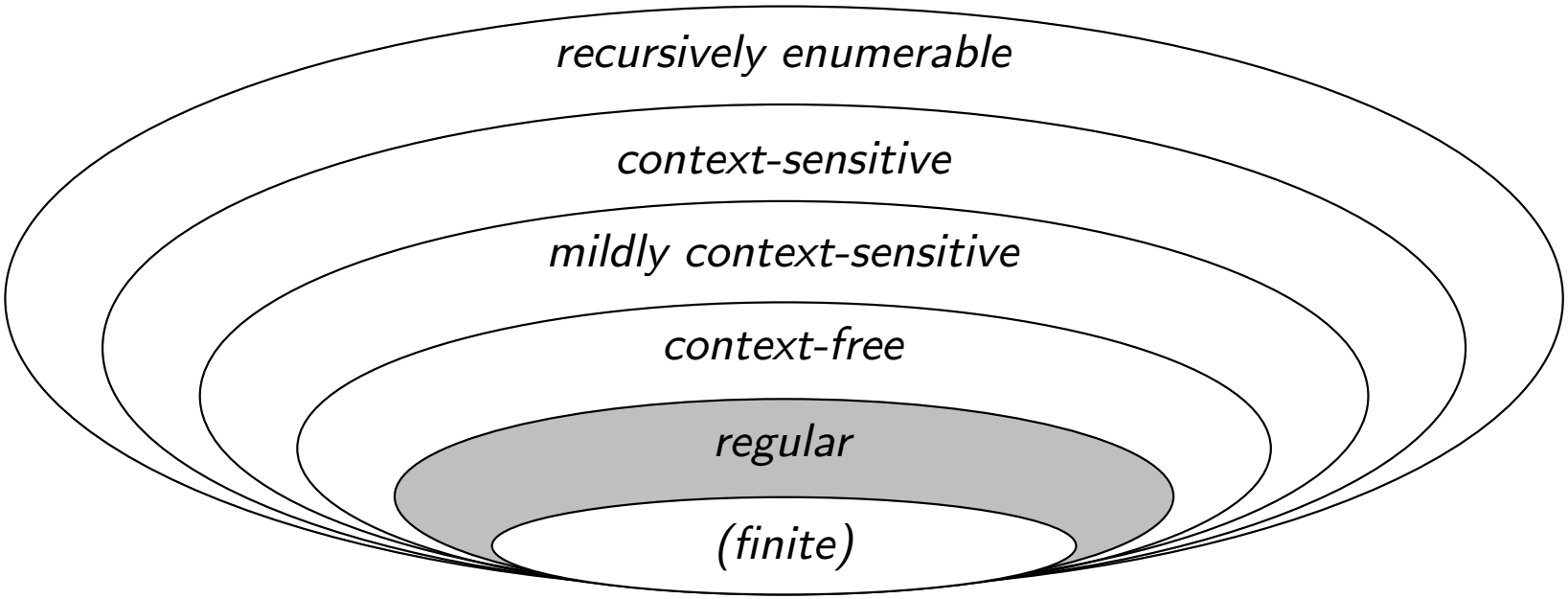
Science

Classifying an object in terms of its complexity helps to study and predict its properties.

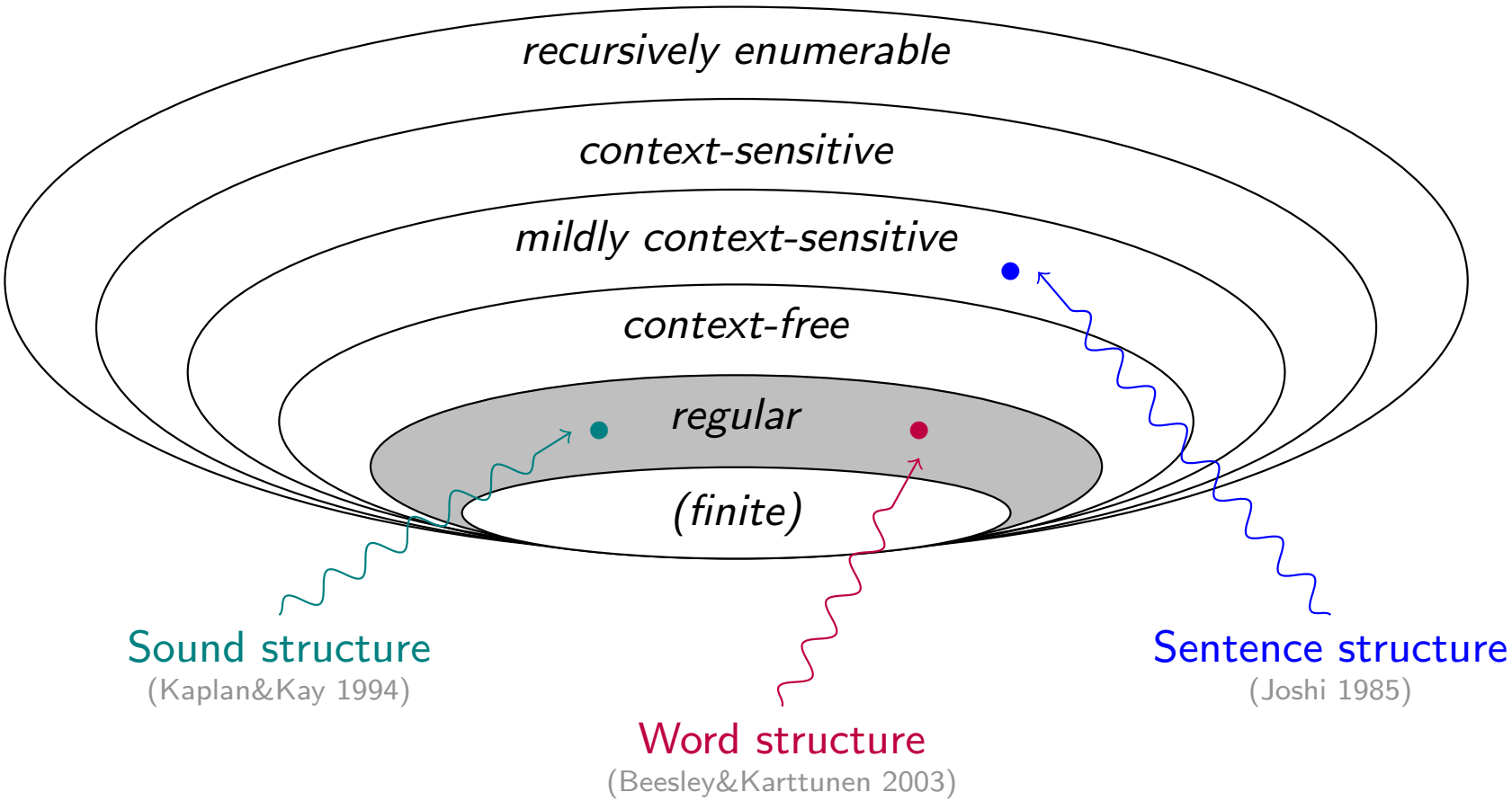
Industry

Knowing types of dependencies, we can design better processing techniques and learners.

The Chomsky Hierarchy of String Languages



The Chomsky Hierarchy of String Languages



Regular languages

Regular languages = FSA recognized = MSO definable

Regular languages

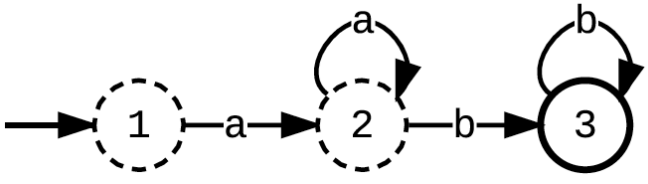
Regular languages = FSA recognized = MSO definable

a^+b^+

Regular languages

Regular languages = FSA recognized = MSO definable

a^+b^+



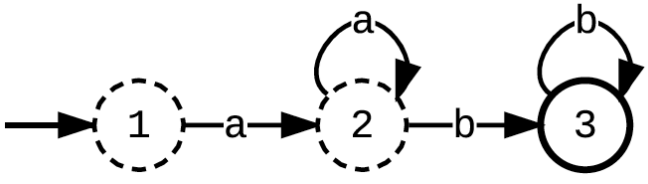
Regular

Regular languages

Regular languages = FSA recognized = MSO definable

a^+b^+

a^nb^n

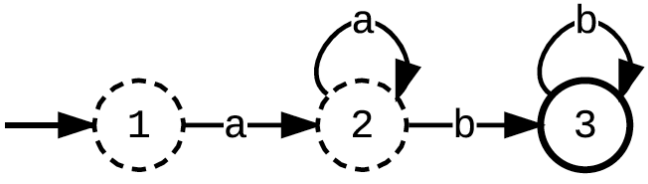


Regular

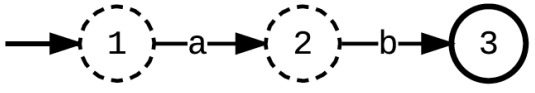
Regular languages

Regular languages = FSA recognized = MSO definable

a^+b^+



a^nb^n

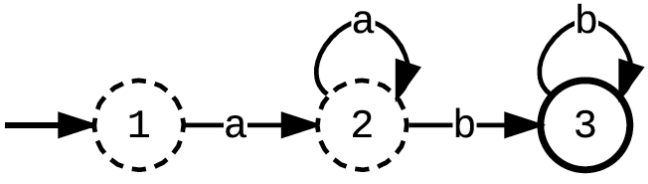


Regular

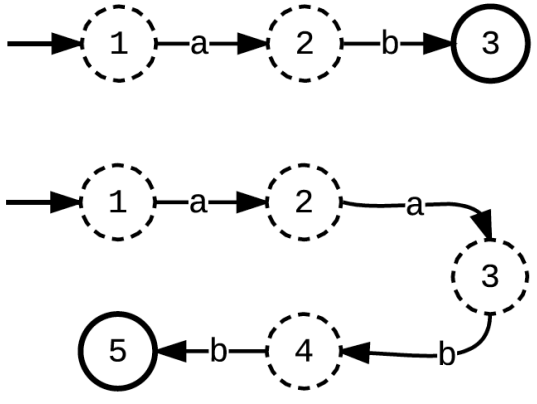
Regular languages

Regular languages = FSA recognized = MSO definable

a^+b^+



a^nb^n

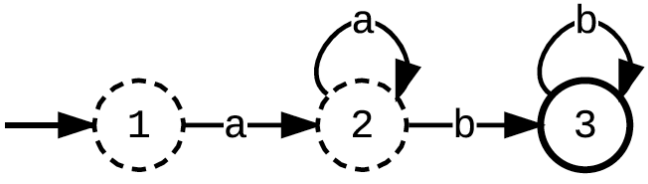


Regular

Regular languages

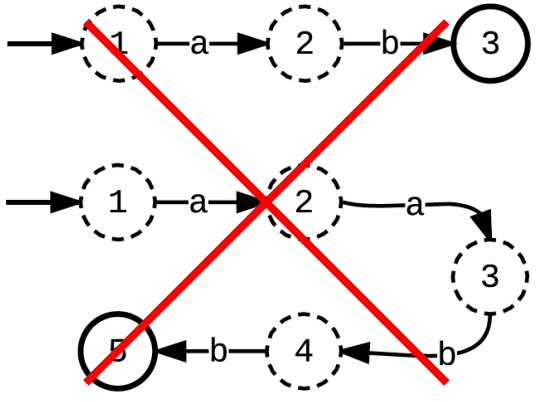
Regular languages = FSA recognized = MSO definable

a^+b^+



Regular

a^nb^n



. . . no general FSA can be constructed

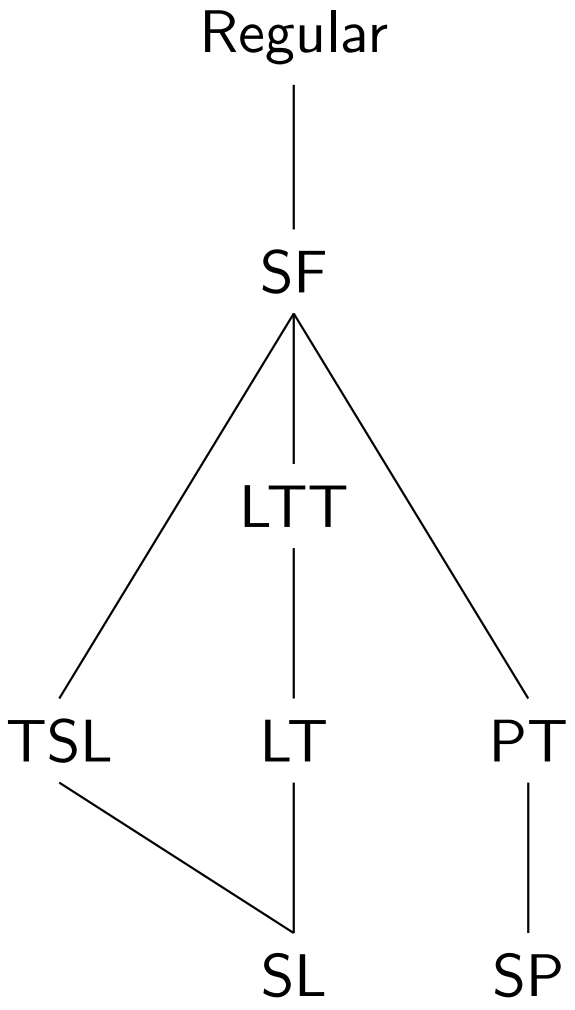
Context-Free

Subregular Hierarchy

The class of regular languages can be decomposed into **subregular hierarchy**

- Introduced by McNaughton&Papert (1971)
- Expanded by numerous researchers

Subregular hierarchy



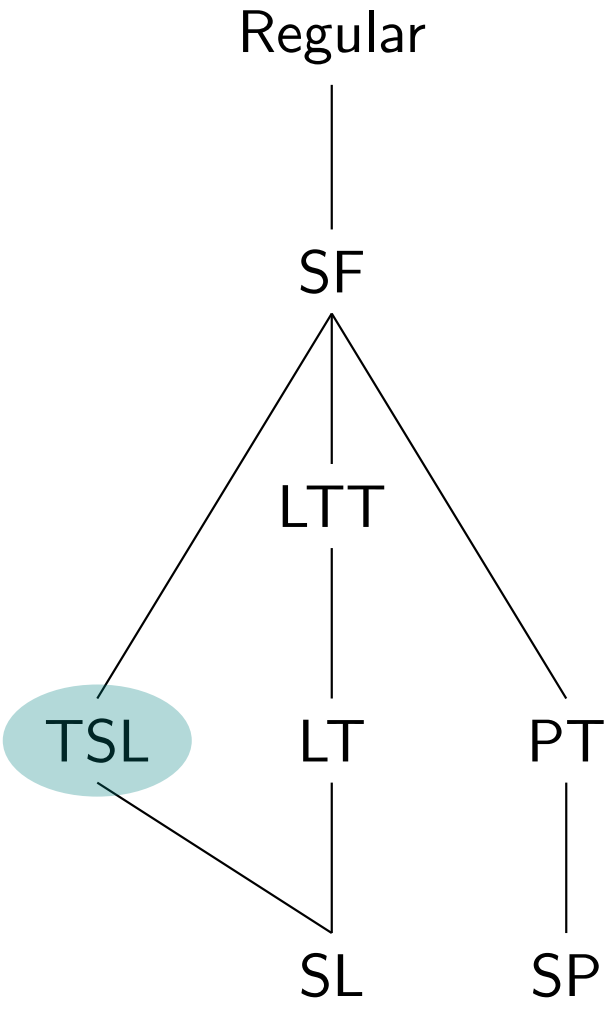
Subregular Hierarchy

The class of regular languages can be decomposed into **subregular hierarchy**

- Introduced by McNaughton&Papert (1971)
- Expanded by numerous researchers

The most fruitful class for the NL is tier-based strictly local languages (TSL).

Subregular hierarchy



TSL intuitions

Intuition: TSL is a *n*-gram model on steroids.

TSL intuitions

Intuition: TSL is a *n*-gram model on steroids.

n-gram model or SL grammar lists the (im)possible sequences of elements.

*ab constraint:

*a a c a b a c

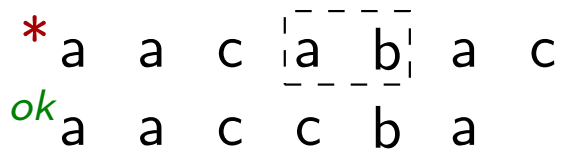
ok a a c c b a

TSL intuitions

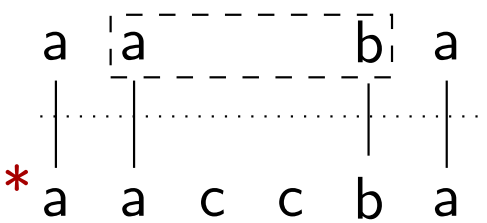
Intuition: TSL is a *n*-gram model on steroids.

n-gram model or SL grammar lists the (im)possible sequences of elements.

*ab constraint:



TSL grammar is a SL grammar of a certain subset of the alphabet (*tier alphabet*). Other elements are ignored.



TSL example: (im)possible iteration

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

TSL example: (im)possible iteration

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

$G = \langle \rtimes\text{-}\ltimes,\ \rtimes\text{-posle},\ \text{posle-}\ltimes,\ \text{posle-posle} \rangle$

TSL example: (im)possible iteration

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

$G = \langle \rtimes\text{-}\ltimes,\ \rtimes\text{-posle},\ \text{posle-}\ltimes,\ \text{posle-posle} \rangle$

Ilocano

2a. bigat
morning

2b. ka-bigat-an
the next morning

2c. $^{*}\text{ka}^{n}\text{-bigat-an}^{n}$
*the morning after the next one

TSL example: (im)possible iteration

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

$G = \langle \rtimes\text{-}\ltimes,\ \rtimes\text{-posle},\ \text{posle-}\ltimes,\ \text{posle-posle} \rangle$

Ilocano

2a. bigat
morning

2b. ka-bigat-an
the next morning

2c. $^{*}\text{ka}^{n}\text{-bigat-an}^{n}$
*the morning after the next one

$G = \langle \rtimes\text{-}\ltimes,\ \rtimes\text{-ka},\ \text{an-}\ltimes,\ \text{ka-an} \rangle$

TSL example: (im)possible iteration

Russian

1a. zavtra
tomorrow

1b. posle-zavtra
the day after tomorrow

1c. posle-posle-zavtra
the day after the day after tomorrow

$G = \langle \rtimes\text{-}\ltimes,\ \rtimes\text{-posle},\ \text{posle-}\ltimes,\ \text{posle-posle} \rangle$

Ilocano

2a. bigat
morning

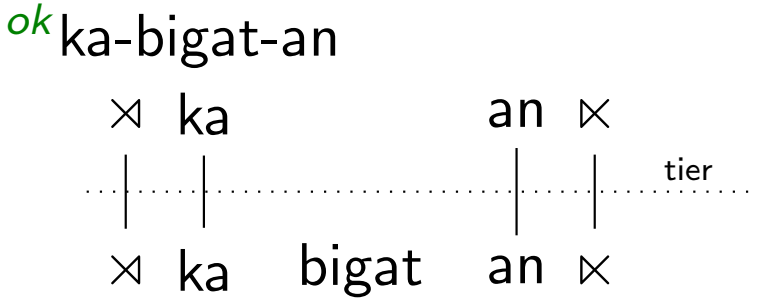
2b. ka-bigat-an
the next morning

2c. $^{*}\text{ka}^{n}\text{-bigat-an}^{n}$
*the morning after the next one

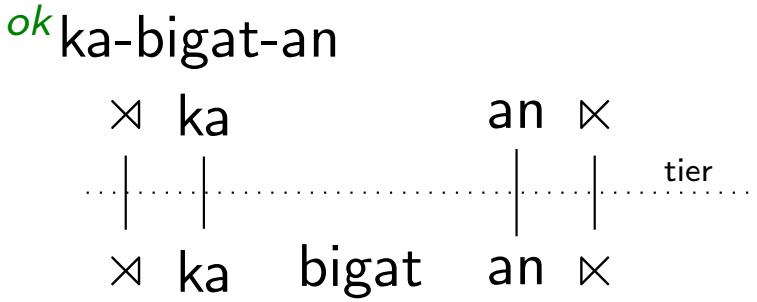
$G = \langle \rtimes\text{-}\ltimes,\ \rtimes\text{-ka},\ \text{an-}\ltimes,\ \text{ka-an} \rangle$

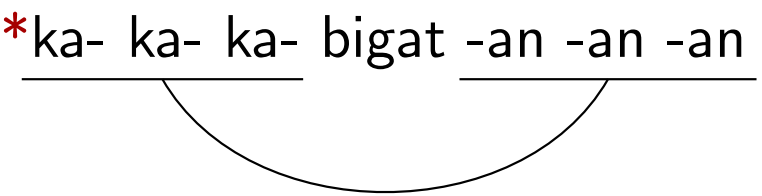
- Russian & Ilocano: radically different, both TSL
- 2c pattern: not TSL, unattested

TSL example: (im)possible iteration [cont.]



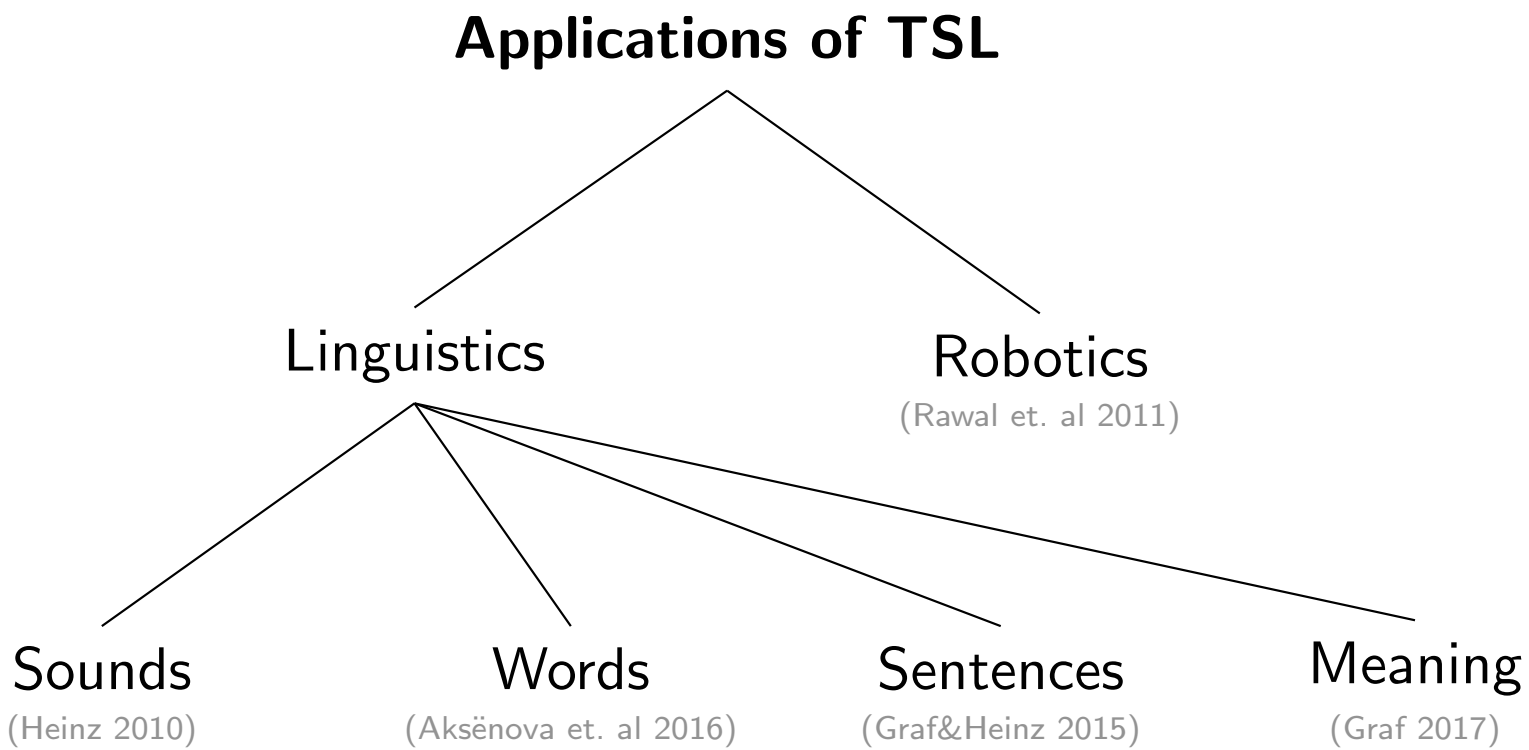
TSL example: (im)possible iteration [cont.]





$a^n b^n$ pattern $\rightarrow$ CFG!

Applications of TSL



Subregular toolkit: motivations

Motivations:

- A tool to avoid the pen-and-paper way to create subregular grammars and generate data samples of a particular type in order to test certain hypothesis.

Subregular toolkit: motivations

Motivations:

- A tool to avoid the pen-and-paper way to create subregular grammars and generate data samples of a particular type in order to test certain hypothesis.
- A band to connect explorations of human "software" and its possible implementation.

Subregular toolkit: motivations

Motivations:

- A tool to avoid the pen-and-paper way to create subregular grammars and generate data samples of a particular type in order to test certain hypothesis.
- A band to connect explorations of human "software" and its possible implementation.
- A way to implement learners for subregular languages and measure their performance.

Subregular toolkit: motivations

Motivations:

- A tool to avoid the pen-and-paper way to create subregular grammars and generate data samples of a particular type in order to test certain hypothesis.
- A band to connect explorations of human "software" and its possible implementation.
- A way to implement learners for subregular languages and measure their performance.
- A possibility to use subregular tools in "real life" for language processing problems.

Subregular toolkit: motivations

Motivations:

- A tool to avoid the pen-and-paper way to create subregular grammars and generate data samples of a particular type in order to test certain hypothesis.
- A band to connect explorations of human "software" and its possible implementation.
- A way to implement learners for subregular languages and measure their performance.
- A possibility to use subregular tools in "real life" for language processing problems.
- An instrument to explore learnability of certain types of NL patterns.

Subregular toolkit: what's in it?

Tools:

- Sample generators
- Scanners
- Learners

Subregular toolkit: what's in it?

Tools:

- Sample generators
- Scanners
- Learners

Language classes (minimum):

- Strictly local
- Tier-based strictly local
  - "standard" definition
  - with structural projection mechanism
  - with multiple tiers
- Strictly piecewise
- ... and other subregular classes

Subregular toolkit: details

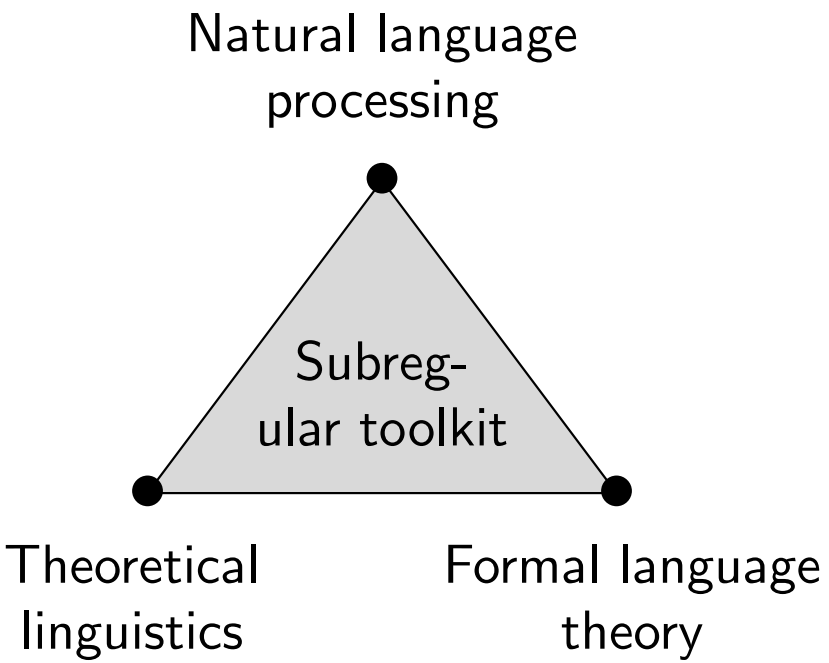
- Python 3 (available via `pip`)
  # Python is widely used in scientific community
- Open source
  # to make it available for further enhancement
- With GUI
  # to simplify the use

- Available on GitHub
  # `https://github.com/loisetoil/slp`

Summary

Subregular toolkit allows:

- test ideas currently available in literature;
- explore new methods to model NL;
- seek out new ways to improve the results.



Thank you!

> [Science] is a system for testing your thoughts against the universe and seeing whether they match.
>
> *Isaac Asimov*

References I

Aksënova, Alëna (2017)

To achieve harmony we only need one tier.
Slides of a talk given at PLC 41. Philadelphia, PA.

Aksënova, Alëna and Aniello De Santo (2017)

Strict Locality in Morphological Derivations.
Slides of a talk given at CLS 53. Chicago, IL.

Aksënova, Alëna and Marina Ermolaeva (2017)

Diretra, a customizable direct translation system: first sketches.
In *Proceedings of TRANSLATA II*. Peter Lang, Germany.

Aksënova, Alëna, Thomas Graf and Sedigheh Moradi (2016)

Morphotactics as Tier-Based Strictly Local Dependencies.
In *Proceedings of SIGMorPhon 2016*.

Beesley, Kenneth and Lauri Kartunnen (2003)

Finite state morphology.
Stanford, CA: CSLI Publications.

Graf, Thomas, Alëna Aksënova and Aniello De Santo (2016)

A Single Movement Normal Form for Minimalist Grammars.
In *Formal Grammar: Lecture Notes in Computer Science, vol. 9804*.

Graf, Thomas and Jeffrey Heinz (2015)

Commonality in Disparity: The Computational View of Syntax and Phonology.
Slides of a talk given at GLOW 2015. Paris, France.

References II

Heinz, Jeffrey (2010)

Learning long-distance phonotactics.
Linguistic Inquiry 41(4): 623 – 661.

Graf, Thomas (2017)

The subregular complexity of monomorphemic quantifiers.
Ms., Stony Brook University.

Joshi, Aravind (1985)

Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?
In *Natural Language Parsing*. Cambridge University Press.

Kaplan, Ronald and Martin Kay (1994)

Regular Models of Phonological Rule Systems.
Computational Linguistics 20(3), 331 – 378.

McNaughton, Robert and Seymour Papert (1971)

Counter-Free Automata.
MIT Press, Cambridge.

Rawal, Chetan, Herbert Tanner and Jeffrey Heinz (2011)

(Sub)regular Robotic Languages.
In *Proceedings of IEEE Mediterranean Conference on Control and Automation*, 321–326.