



# Strict Locality In Morphological Derivations

Alëna Aksënova and Aniello De Santo

Stony Brook University

CLS 53

The University of Chicago

May 26, 2017

# Motivations

Our goal is to give arguments towards derivational representations in morphology.

## Idea

We can exploit Formal Language Theory to

- abstract from narrow, framework-specific details;
- quantify theoretical intuitions;
- *bonus*: cross-domain complexity parallels.

## Spoilers:

- long-distance dependencies can be viewed as local
- descriptions of the patterns are much more succinct
- this results in a reduced computational complexity

# Motivations

Our goal is to give arguments towards derivational representations in morphology.

## Idea

We can exploit Formal Language Theory to

- abstract from narrow, framework-specific details;
- quantify theoretical intuitions;
- *bonus*: cross-domain complexity parallels.

## Spoilers:

- long-distance dependencies can be viewed as local
- descriptions of the patterns are much more succinct
- this results in a reduced computational complexity

# Motivations

Our goal is to give arguments towards derivational representations in morphology.

## Idea

We can exploit Formal Language Theory to

- abstract from narrow, framework-specific details;
- quantify theoretical intuitions;
- *bonus*: cross-domain complexity parallels.

## Spoilers:

- long-distance dependencies can be viewed as local
- descriptions of the patterns are much more succinct
- this results in a reduced computational complexity

# Outline

- 1 Derived vs. Derivational
- 2 Subregular Morphology
- 3 Russian Nominalization
- 4 SL Derivations
- 5 Conclusion

# Morphological representations

## How to evaluate the grammaticality of morphological forms?

- **Derived sequences:** evaluating the resulting sequence after all operations were applied.  
⇒ Under this perspective, morphology is not hierarchical, it is simply concatenation of smaller strings.  
(McGregor 2003), i.a.
- **Derivational sequences:** instead of looking at the output, considering the operations that were applied to the root node.  
⇒ Under this perspective, morphology is hierarchical, and the order in which the affixes were applied matters.  
(Vikner&Vikner 2008), i.a.

# Morphological representations

## How to evaluate the grammaticality of morphological forms?

- **Derived sequences:** evaluating the resulting sequence after all operations were applied.  
⇒ Under this perspective, morphology is not hierarchical, it is simply concatenation of smaller strings.  
(McGregor 2003), i.a.
- **Derivational sequences:** instead of looking at the output, considering the operations that were applied to the root node.  
⇒ Under this perspective, morphology is hierarchical, and the order in which the affixes were applied matters.  
(Vikner&Vikner 2008), i.a.

# Morphological representations

## How to evaluate the grammaticality of morphological forms?

- **Derived sequences:** evaluating the resulting sequence after all operations were applied.  
⇒ Under this perspective, morphology is not hierarchical, it is simply concatenation of smaller strings.  
(McGregor 2003), i.a.
- **Derivational sequences:** instead of looking at the output, considering the operations that were applied to the root node.  
⇒ Under this perspective, morphology is hierarchical, and the order in which the affixes were applied matters.  
(Vikner&Vikner 2008), i.a.

# Derived vs. derivational approaches: an example

## un-lock-able

### Derived approach

1. Semantics is extracted based on the form of the string
2. Representation:  
un + lock + able

### Derivational approach

1. Semantics is extracted based on the order of applied operations
2. Representation:  
[lock + able] + un  
[lock + un] + able

# Derived vs. derivational approaches: an example

## un-lock-able

### Derived approach

1. Semantics is extracted based on the form of the string
2. Representation:  
un + lock + able

### Derivational approach

1. Semantics is extracted based on the order of applied operations
2. Representation:  
[lock + able] + un  
[lock + un] + able

# Derived vs. derivational approaches: an example

## un-lock-able

### Derived approach

1. Semantics is extracted based on the form of the string
2. Representation:  
un + lock + able

### Derivational approach

1. Semantics is extracted based on the order of applied operations
2. Representation:  
[lock + able] + un  
[lock + un] + able

# One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.



Can we approach this problem from a more formal point of view?

# One topic of debate: Semantic ambiguity

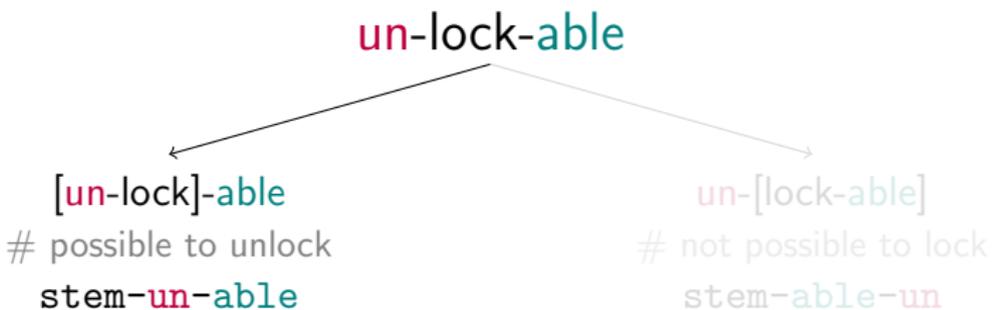
The derivational representation captures the semantic ambiguity caused by different order of affix application.



Can we approach this problem from a more formal point of view?

# One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.

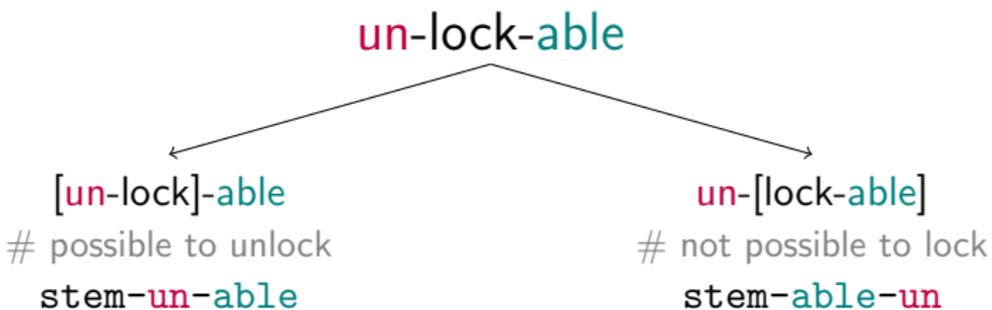


Can we approach this problem from a more formal point of view?



# One topic of debate: Semantic ambiguity

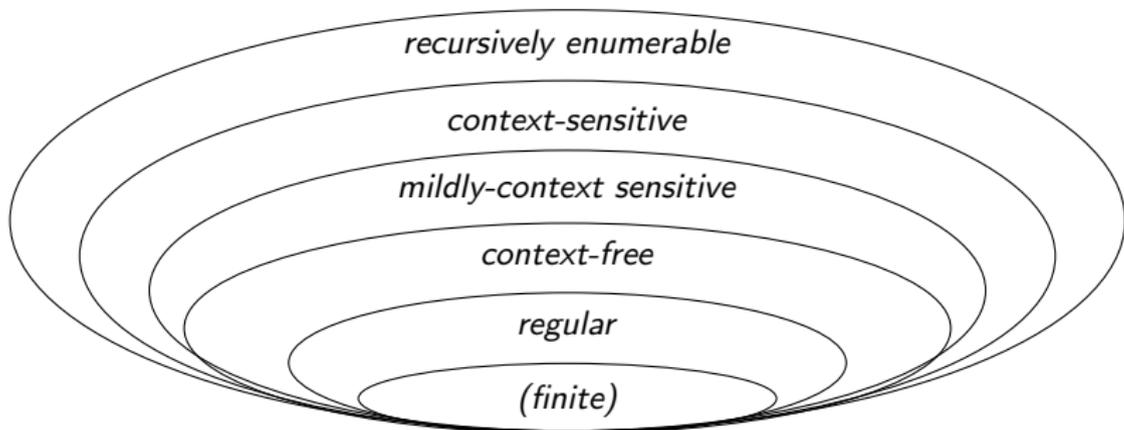
The derivational representation captures the semantic ambiguity caused by different order of affix application.



Can we approach this problem from a more formal point of view?

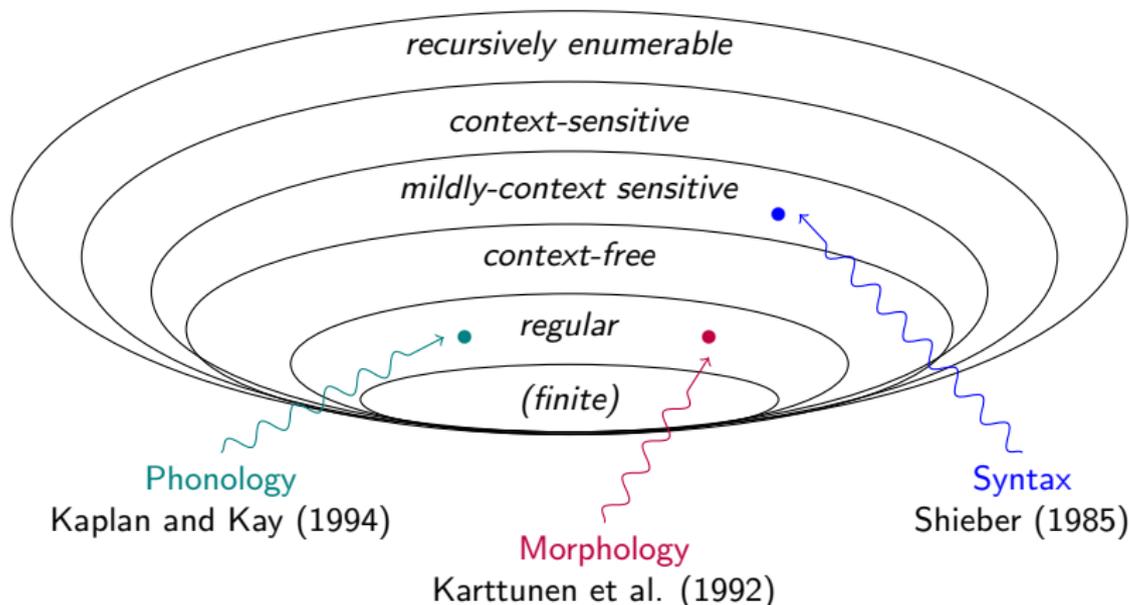
# The Chomsky Hierarchy of String Languages

Languages (stringsets) can be classified according to the complexity of the grammars that generate them.

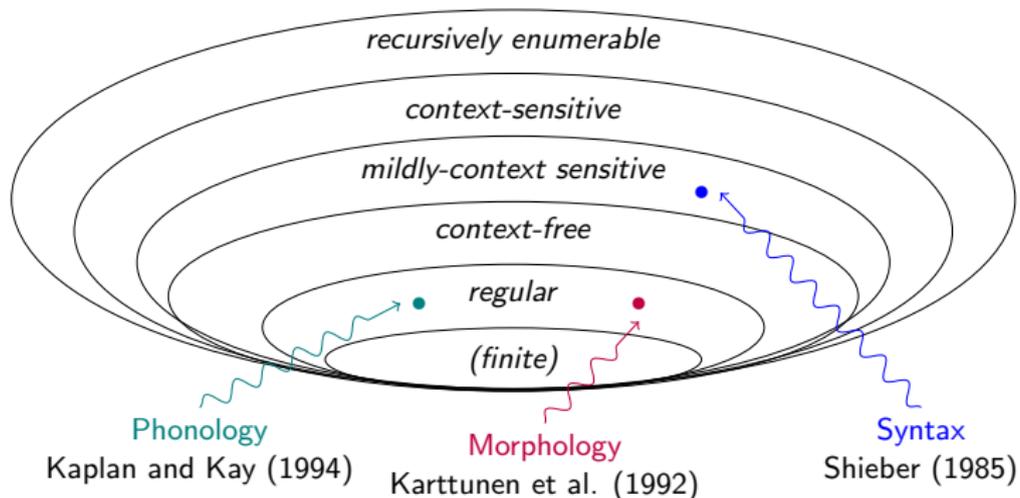


# The Chomsky Hierarchy of String Languages

Languages (stringsets) can be classified according to the complexity of the grammars that generate them.



# Morphology as a Regular System



## Precise predictions for:

- typology → e.g. no unbounded center embedding
- learnability → e.g. no Gold learning for regular languages
- cognition → e.g. finitely bounded working memory

# Subregular Hierarchy

Not full power of finite machinery is needed

⇒ **subregular hierarchy**

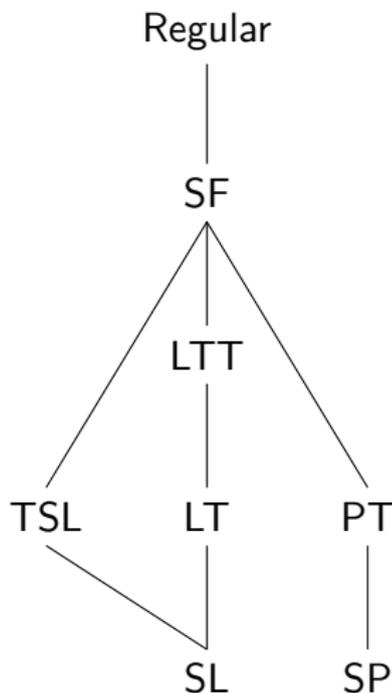
- Subregular hierarchy introduced (McNaughton&Papert 1971)
- Subregular hierarchy expanded (Rogers et. al 2010)
- Phonology is subregular (Heinz&Iidsardi 2013)
- Morphotactics is subregular (Aksënova et. al 2016)

## Subregular Morphotactics

**Morphotactic** dependencies seem to be

- strictly local (SL)
- tier-based strictly local (TSL)

## Subregular hierarchy



# Subregular Hierarchy

Not full power of finite machinery is needed

⇒ **subregular hierarchy**

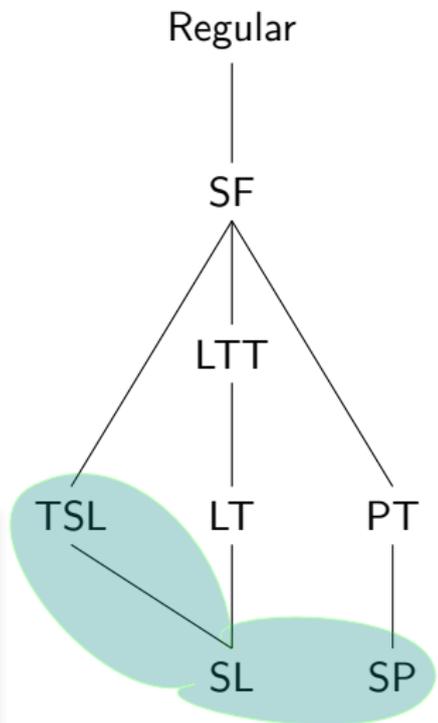
- Subregular hierarchy introduced (McNaughton&Papert 1971)
- Subregular hierarchy expanded (Rogers et. al 2010)
- **Phonology** is subregular (Heinz&Idsardi 2013)
- **Morphotactics** is subregular (Aksënova et. al 2016)

## Subregular Morphotactics

**Morphotactic** dependencies seem to be

- strictly local (SL)
- tier-based strictly local (TSL)

## Subregular hierarchy



# Subregular Hierarchy

Not full power of finite machinery is needed

⇒ **subregular hierarchy**

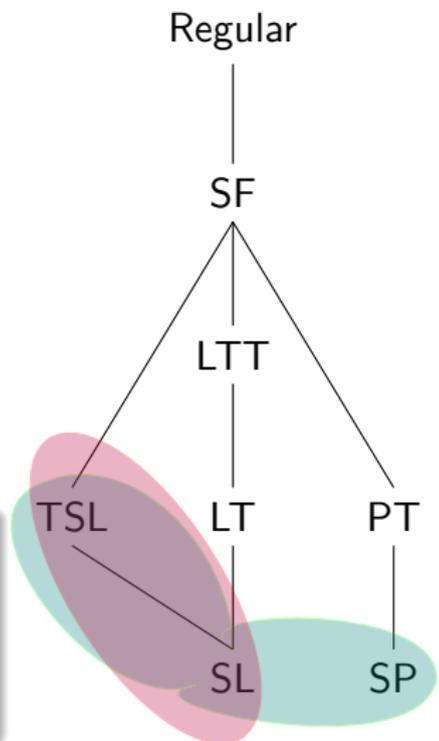
- Subregular hierarchy introduced (McNaughton&Papert 1971)
- Subregular hierarchy expanded (Rogers et. al 2010)
- **Phonology** is subregular (Heinz&Idsardi 2013)
- **Morphotactics** is subregular (Aksënova et. al 2016)

## Subregular Morphotactics

**Morphotactic** dependencies seem to be

- strictly local (SL)
- tier-based strictly local (TSL)

## Subregular hierarchy



# SL morphotactics: affixation

**Strictly local (SL) grammars** capture *local* dependencies by listing **disallowed substrings**.

## Example (Affixation in English)

- *un-* is a prefix: *un-holy*, *un-do*
- *-able* is a suffix: *drink-able*, *move-able*
- $G_{SL} = \{*\text{able-stem}, *\text{stem-un}\}$ 
  - blocks improper ordering
  - predicts co-occurrence of these affixes
- Indeed, it is correct:
  - *ok*do, *ok*un-do, *ok*un-do-able
  - *\*able-move*, *\*able-do-un*

# SL morphotactics: affixation

**Strictly local (SL) grammars** capture *local* dependencies by listing **disallowed substrings**.

## Example (Affixation in English)

- **un-** is a prefix: **un**-holy, **un**-do
- **-able** is a suffix: drink-**able**, move-**able**
- $G_{SL} = \{*\text{able-stem}, *\text{stem-un}\}$ 
  - blocks improper ordering
  - predicts co-occurrence of these affixes
- Indeed, it is correct:
  - *ok* do, *ok* un-do, *ok* un-do-able
  - *\*able*-move, *\*able*-do-un

# SL morphotactics: affixation

**Strictly local (SL) grammars** capture *local* dependencies by listing **disallowed substrings**.

## Example (Affixation in English)

- **un-** is a prefix: **un**-holy, **un**-do
- **-able** is a suffix: drink-**able**, move-**able**
- $G_{SL} = \{*\mathbf{able-stem}, *\mathbf{stem-un}\}$ 
  - blocks improper ordering
  - predicts co-occurrence of these affixes
- Indeed, it is correct:
  - *ok* do, *ok* **un**-do, *ok* **un**-do-**able**
  - **\*able**-move, **\*able**-do-**un**

# SL morphotactics: affixation

**Strictly local (SL) grammars** capture *local* dependencies by listing **disallowed substrings**.

## Example (Affixation in English)

- **un-** is a prefix: **un**-holy, **un**-do
- **-able** is a suffix: drink-**able**, move-**able**
- $G_{SL} = \{*\mathbf{able-stem}, *\mathbf{stem-un}\}$ 
  - blocks improper ordering
  - predicts co-occurrence of these affixes
- Indeed, it is correct:
  - *ok* do, *ok* **un**-do, *ok* **un**-do-**able**
  - **\*able**-move, **\*able**-do-**un**

# Subregular morphotactics

- The **Subregular approach** abstracts from theory internal details ...
- ... while providing a precise notion of complexity of a certain pattern.

Are there complexity differences among different representations (eg., derived vs. derivational) of the same pattern?

# Subregular morphotactics

- The **Subregular approach** abstracts from theory internal details ...
- ... while providing a precise notion of complexity of a certain pattern.

Are there complexity differences among different representations (eg., derived vs. derivational) of the same patten?

# A case study from Russian

## Russian aspectual metamorphosis

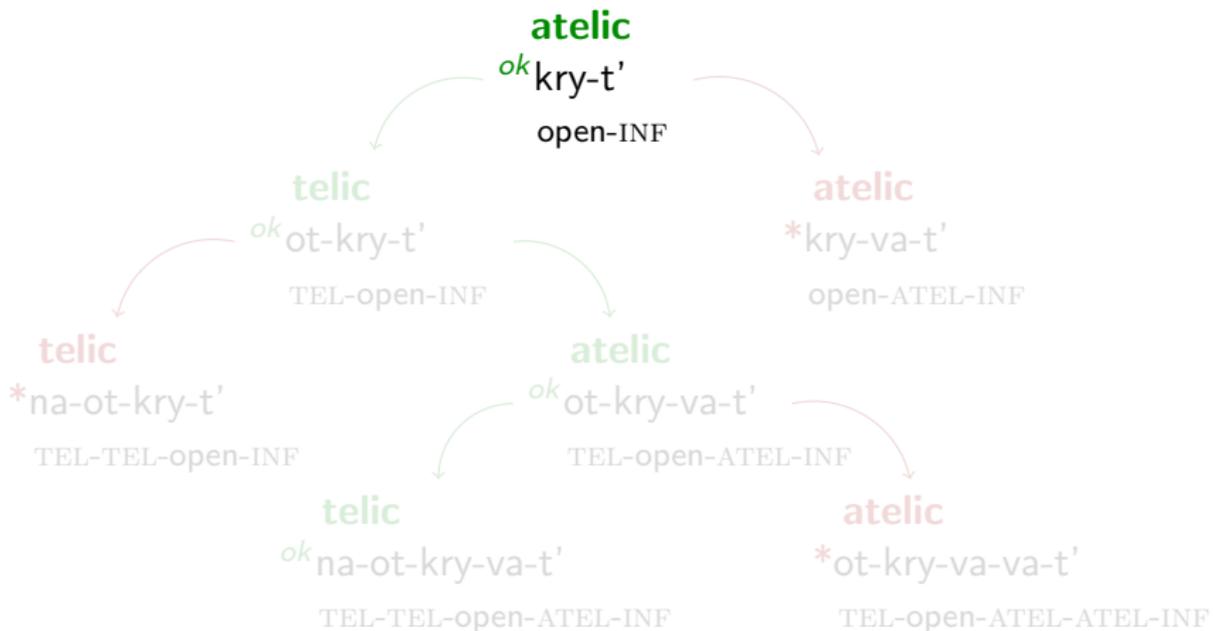
- Stems are intrinsically atelic;
- *telic* prefixes and *atelic* suffix;
- telic prefix can be added only to the atelic form;
- atelic suffix can be added only to the telic form.

## Russian nominalization

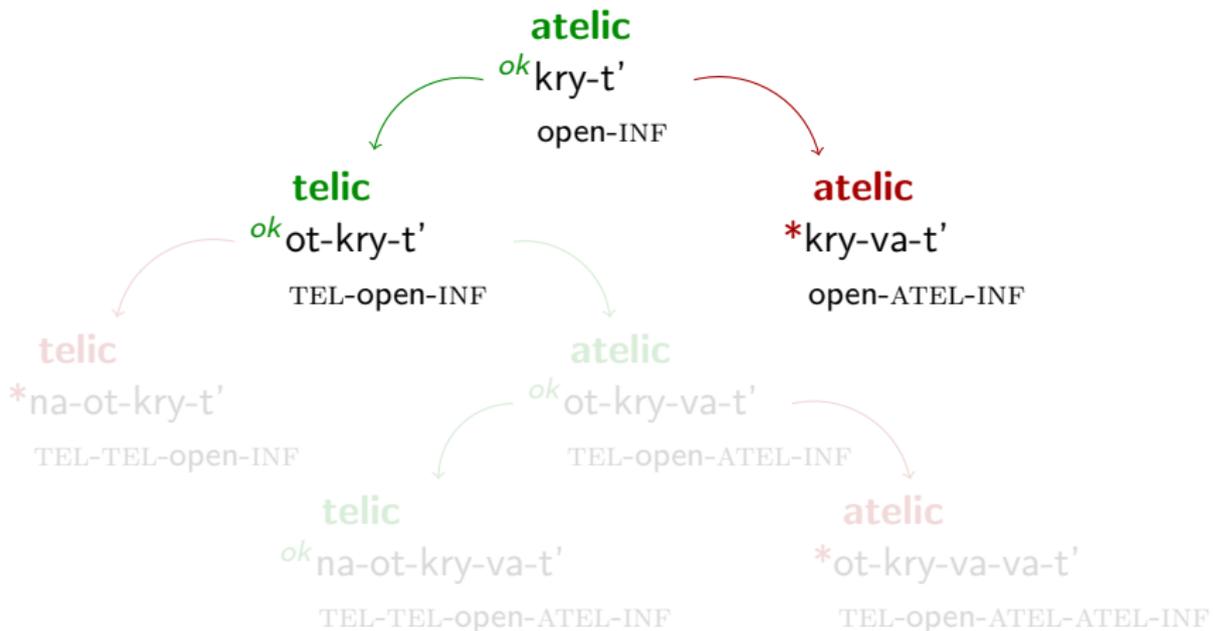
The nominalization suffix:

- cannot apply directly to the stem;
- cannot apply to a telic form;
- can only be applied after the stem is converted to an atelic form.

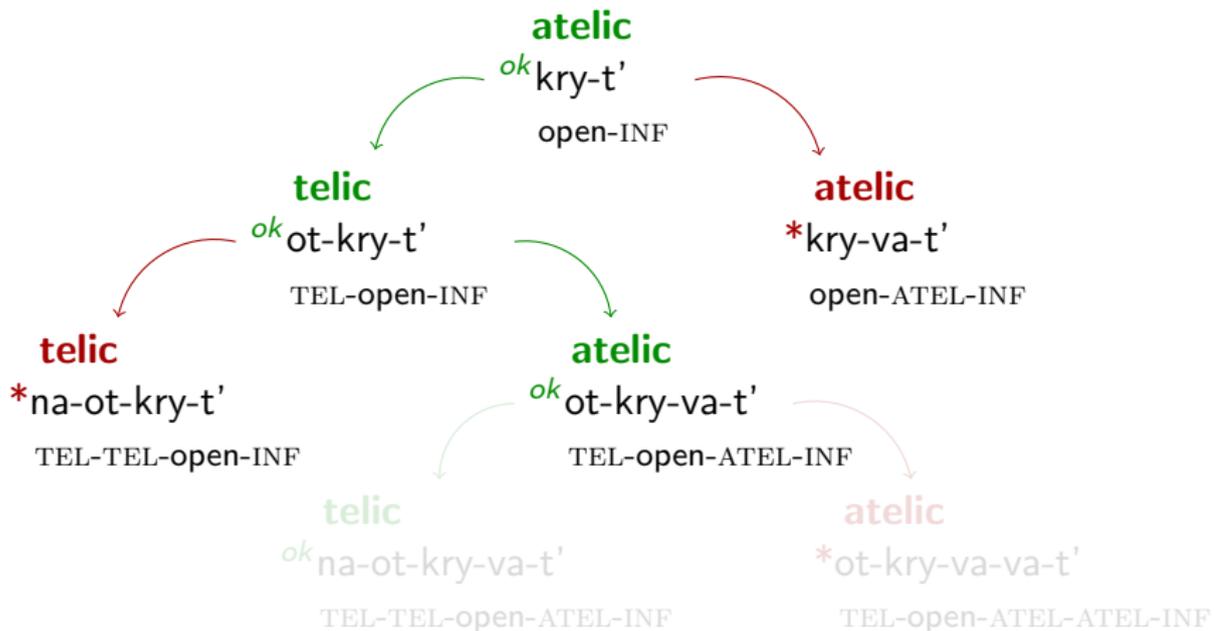
# Russian aspectual metamorphosis



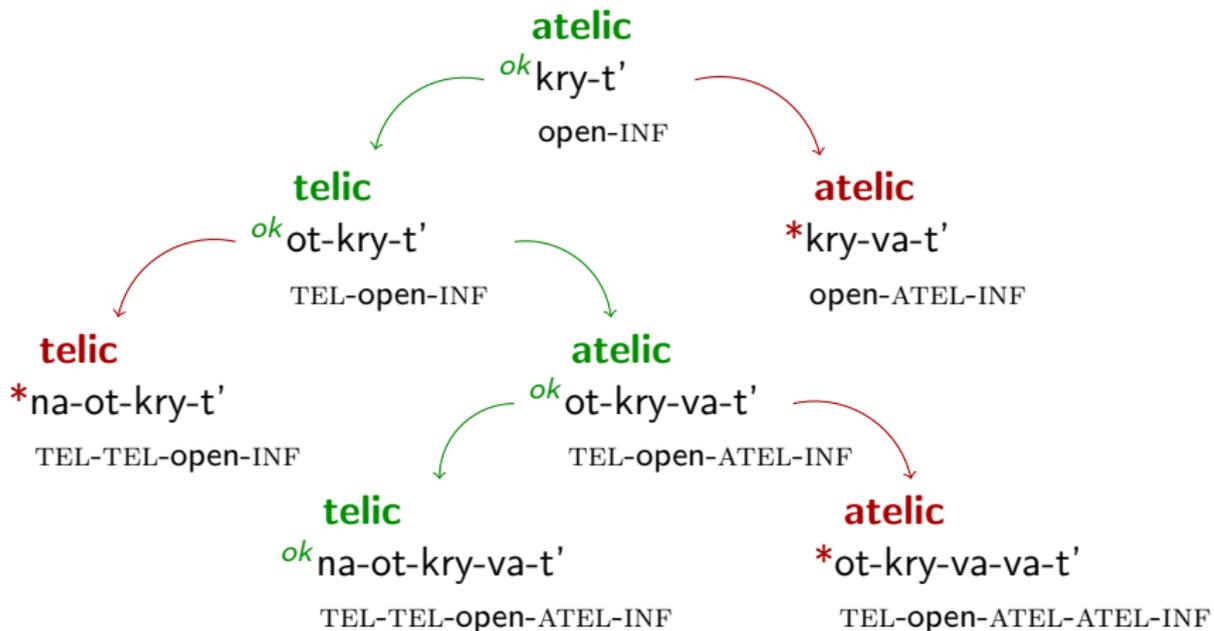
# Russian aspectual metamorphosis



# Russian aspectual metamorphosis



# Russian aspectual metamorphosis



# A case study from Russian

## Russian aspectual metamorphosis

- Stems are intrinsically atelic;
- *telic* prefixes and *atelic* suffixes;
- telic prefix can be added only to the atelic form;
- atelic suffix can be added only to the telic form.

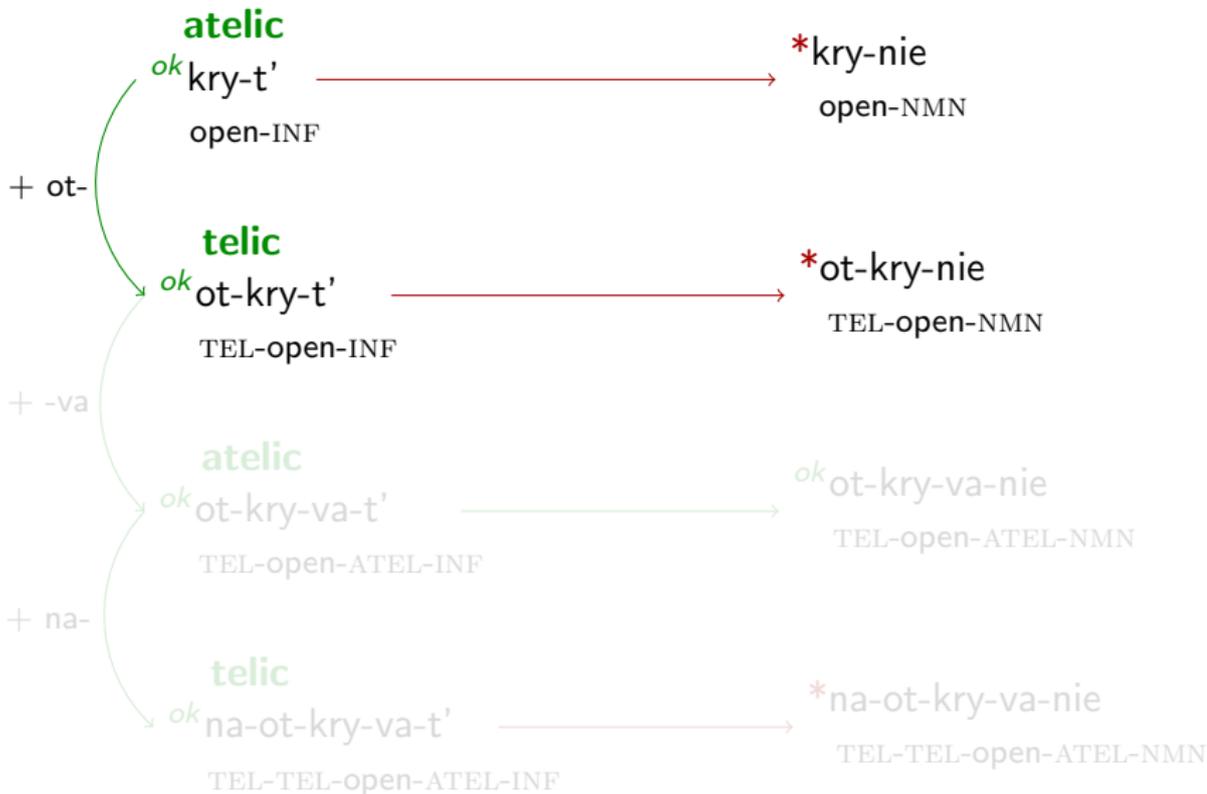
## Russian nominalization

The nominalization suffix:

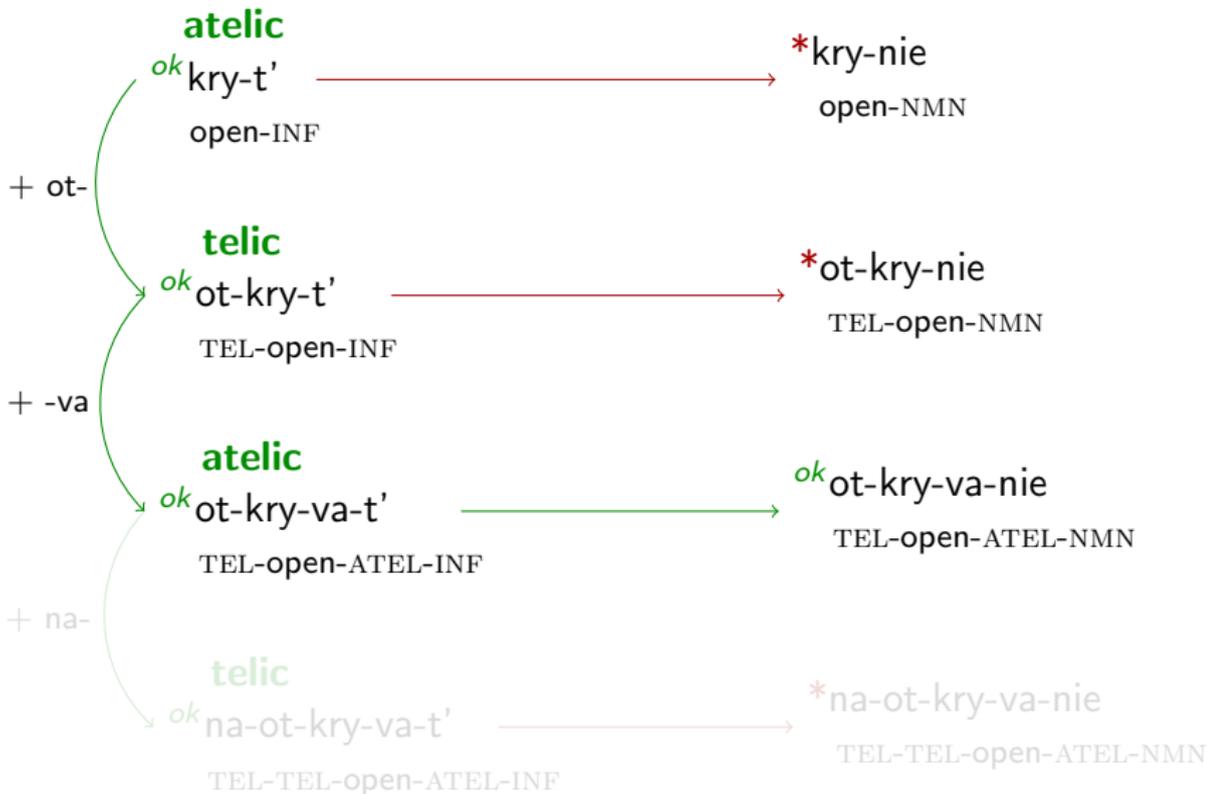
- cannot apply directly to the stem;
- cannot apply to a telic form;
- can only be applied after the stem is converted to an atelic form.

# Russian *-nie* nominalization

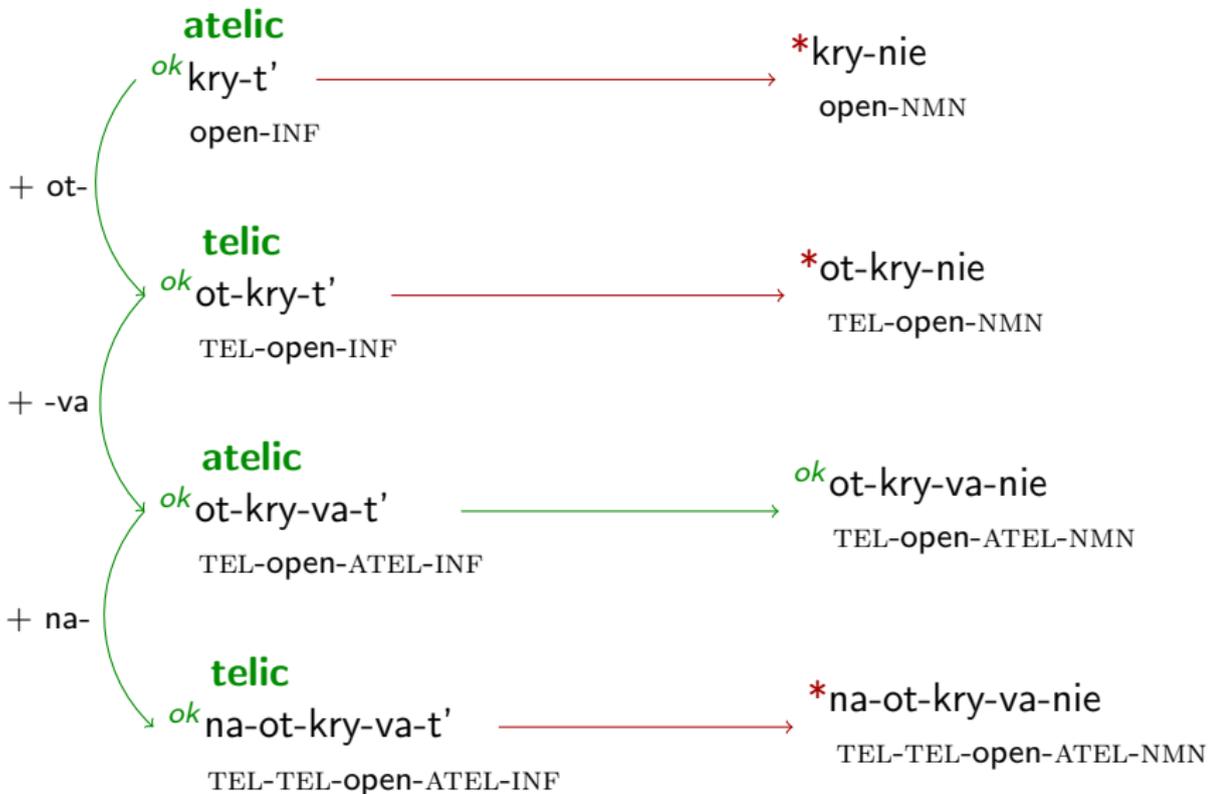


Russian *-nie* nominalization

# Russian *-nie* nominalization



# Russian *-nie* nominalization



## A case study from Russian: summary

### Russian aspectual metamorphosis

- telic prefix can be added only to the atelic form;
- atelic suffix can be added only to the telic form.

### Russian nominalization

- The nominalization suffix can only be applied after the stem is converted to an atelic form.

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-open-ATEL-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

open-ATEL-INF

**telic**

\*na-ot-kry-t'

TEL-TEL-open-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-open-ATEL-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

open-ATEL-INF

**telic**

\*na-ot-kry-t'

TEL-TEL-open-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-~~open-ATEL~~-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

~~open-ATEL~~-INF

**telic**

\*na-ot-kry-t'

TEL-TEL-open-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-open-ATEL-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

open-ATEL-INF

**telic**

\*na-ot-kry-t'

TEL-TEL-open-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-open-ATEL-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

open-ATEL-INF

**telic**

\*na-ot-kry-t'

TEL-TEL-open-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

\*kry-va-t'

~~open-A TEL~~-INF

**atelic**

*ok* ot-kry-va-t'

~~TEL-open-A TEL~~-INF

**telic**

\*na-ot-kry-t'

~~TEL-TEL-open~~-INF

**telic**

*ok* na-ot-kry-va-t'

~~TEL-TEL-open~~-A TEL-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-A TEL-A TEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-open-ATEL-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

open-ATEL-INF

**telic**

\*na-ot-kry-t'

TEL-TEL-open-INF

**telic**

\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

# Russian aspectual metamorphosis: a SL account?

**atelic**

*ok* kry-t'

open-INF

**telic**

*ok* ot-kry-t'

TEL-open-INF

**atelic**

*ok* ot-kry-va-t'

TEL-open-ATEL-INF

**telic**

*ok* na-ot-kry-va-t'

TEL-TEL-open-ATEL-INF

**atelic**

\*kry-va-t'

open-ATEL-INF

**telic**

\*na-ot-kry-t'

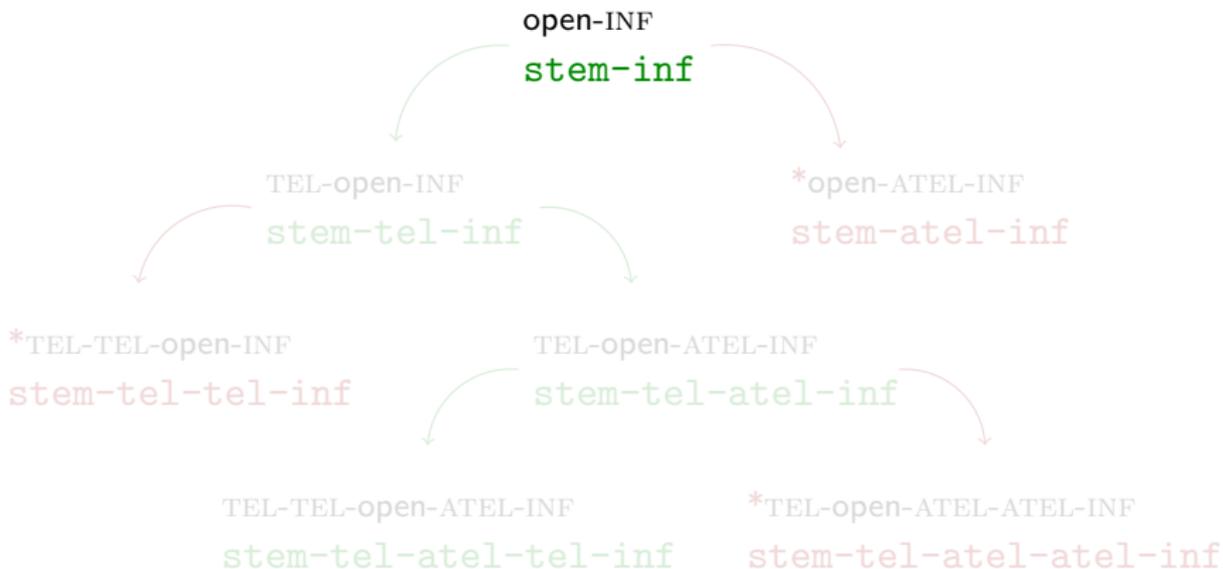
TEL-TEL-open-INF

**telic**

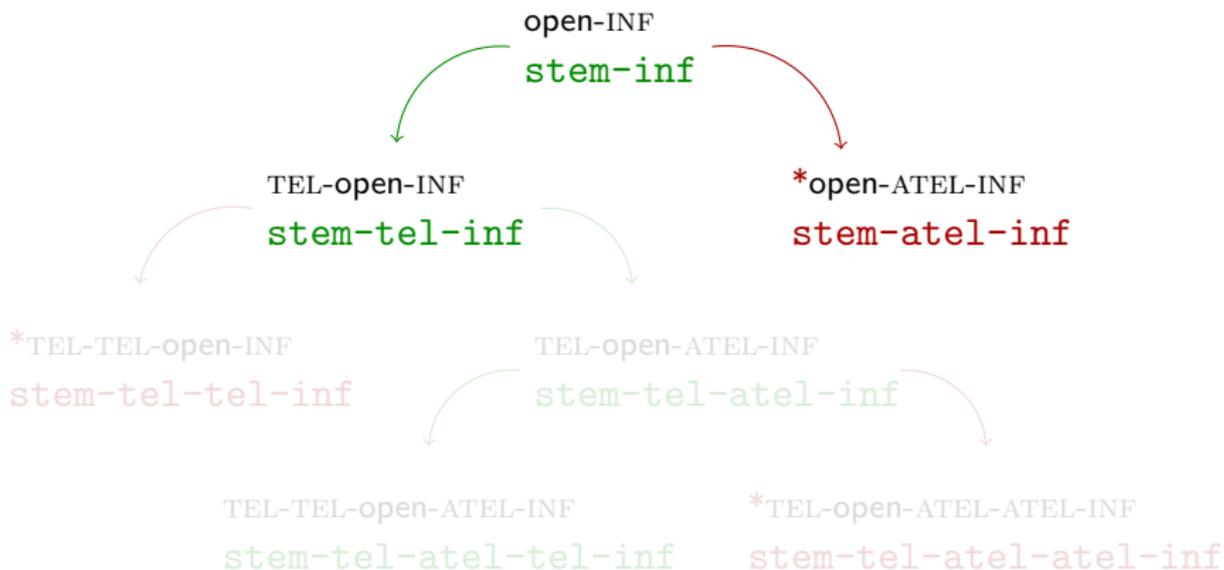
\*ot-kry-va-va-t'

TEL-open-ATEL-ATEL-INF

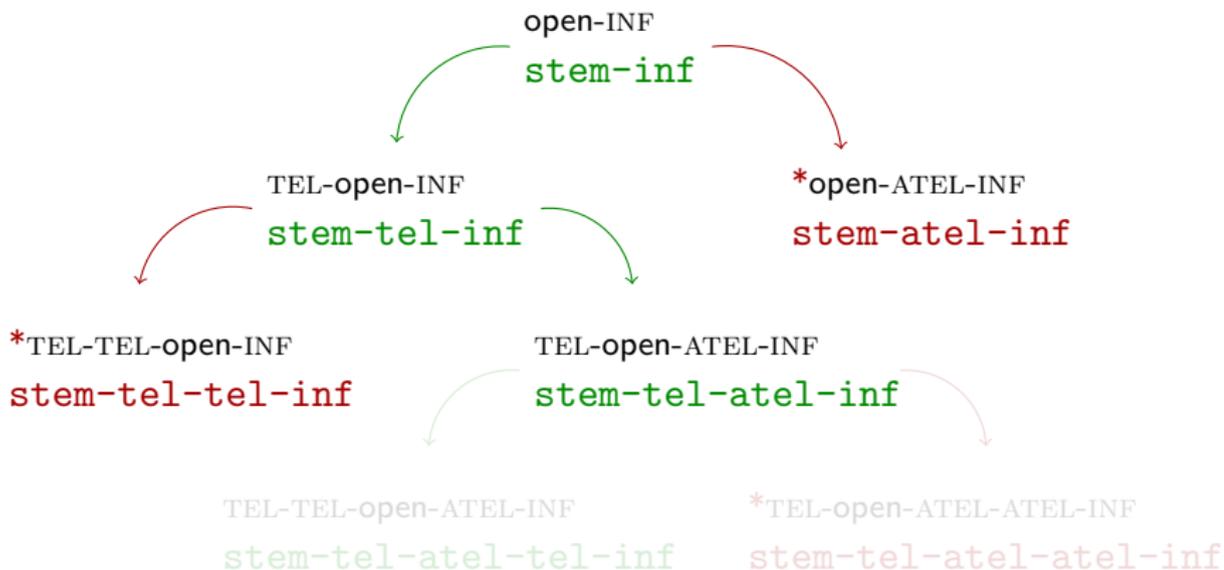
# Russian aspectual metamorphosis: derivational approach



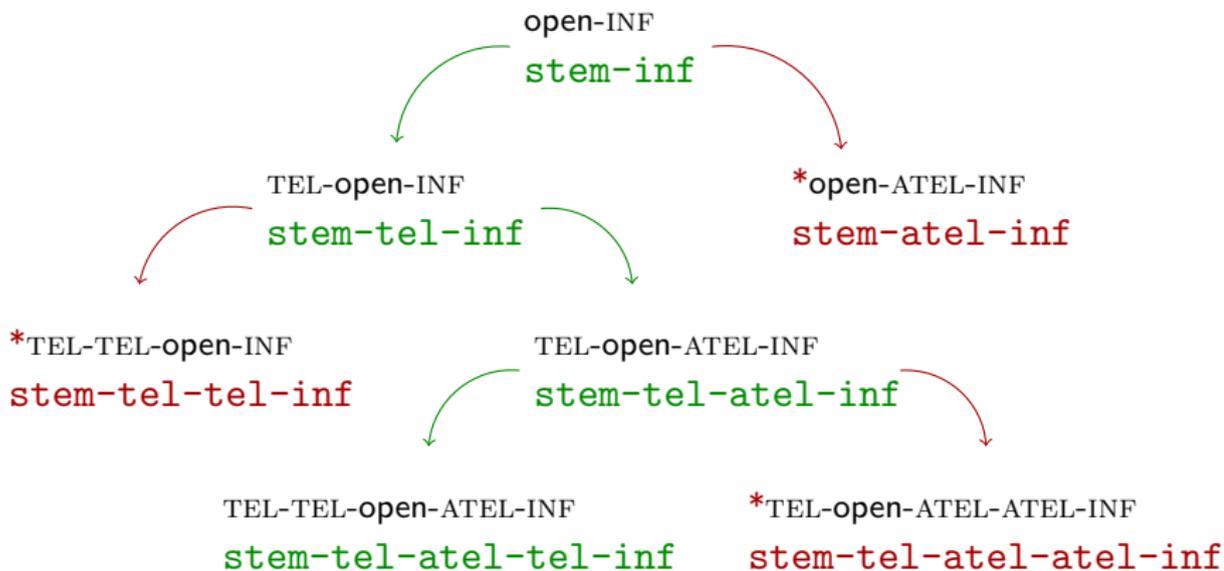
# Russian aspectual metamorphosis: derivational approach



# Russian aspectual metamorphosis: derivational approach



# Russian aspectual metamorphosis: derivational approach



# Russian aspectual pattern: SL derivations

- \*stem-atel: do not add the atelic suffix to the verbal stem;

\*kry-va-t'

stem-atel-inf

- \*tel-tel: ban telic prefix if it was added right before it;

\*na-ot-kry-t'

stem-tel-tel-inf

- \*atel-atel: ban atelic suffix if it was added right before it.

\*ot-kry-va-va-t'

stem-tel-atel-atel-inf

# Russian aspectual pattern: SL derivations

- \*stem-atel: do not add the atelic suffix to the verbal stem;

\*kry-va-t'

stem-atel-inf

- \*tel-tel: ban telic prefix if it was added right before it;

\*na-ot-kry-t'

stem-tel-tel-inf

- \*atel-atel: ban atelic suffix if it was added right before it.

\*ot-kry-va-va-t'

stem-tel-atel-atel-inf

# Russian aspectual pattern: SL derivations

- \*stem-atel: do not add the atelic suffix to the verbal stem;

\*kry-va-t'

stem-atel-inf

- \*tel-tel: ban telic prefix if it was added right before it;

\*na-ot-kry-t'

stem-tel-tel-inf

- \*atel-atel: ban atelic suffix if it was added right before it.

\*ot-kry-va-va-t'

stem-tel-atel-atel-inf

# Derived vs. derivational representations

<sup>ok</sup> **na-ot-kry-va-t'**

TEL-TEL-open-ATEL-INF

stem-tel-atel-tel-inf

na - {ot - kry - va - t'}

{ot - kry - va - va} - t'

**\*ot-kry-va-va-t'**

TEL-open-ATEL-ATEL-INF

stem-tel-atel-atel-inf

stem - tel - {atel - tel} - inf

stem - tel - {atel - atel} - inf

**Derived strings: 4-SL**

**Derivational strings: 2-SL**

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.

# Derived vs. derivational representations

<sup>ok</sup> na-ot-kry-va-t'

TEL-TEL-open-A TEL-INF

stem-tel-atel-tel-inf

na - [ot - kry - va - t']

[ot - kry - va - va] - t'

\*ot-kry-va-va-t'

TEL-open-A TEL-A TEL-INF

stem-tel-atel-atel-inf

stem - tel - [atel - tel] - inf

stem - tel - [atel - atel] - inf

Derived strings: 4-SL

Derivational strings: 2-SL

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.

# Derived vs. derivational representations

<sup>ok</sup> na-ot-kry-va-t'

TEL-TEL-open-A TEL-INF

stem-tel-atel-tel-inf

na - [ot - kry - va - t']

[ot - kry - va - va] - t'

**Derived strings: 4-SL**

\*ot-kry-va-va-t'

TEL-open-A TEL-A TEL-INF

stem-tel-atel-atel-inf

stem - tel - [atel - tel] - inf

stem - tel - [atel - atel] - inf

**Derivational strings: 2-SL**

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.

# Derived vs. derivational representations

<sup>ok</sup> na-ot-kry-va-t'

TEL-TEL-open-A TEL-INF

stem-tel-atel-tel-inf

na - [ot - kry - va - t']

[ot - kry - va - va] - t'

\*ot-kry-va-va-t'

TEL-open-A TEL-A TEL-INF

stem-tel-atel-atel-inf

stem - tel - [atel - tel] - inf

stem - tel - [atel - atel] - inf

**Derived strings: 4-SL**

**Derivational strings: 2-SL**

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.

# Derived vs. derivational representations

<sup>ok</sup> na-ot-kry-va-t'

TEL-TEL-open-A TEL-INF

stem-tel-atel-tel-inf

na - [ot - kry - va - t']

[ot - kry - va - va] - t'

\*ot-kry-va-va-t'

TEL-open-A TEL-A TEL-INF

stem-tel-atel-atel-inf

stem - tel - [atel - tel] - inf

stem - tel - [atel - atel] - inf

**Derived strings: 4-SL**

**Derivational strings: 2-SL**

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.

# Derived vs. derivational representations

<sup>ok</sup> na-ot-kry-va-t'

TEL-TEL-open-A TEL-INF

stem-tel-atel-tel-inf

na - [ot - kry - va - t']

[ot - kry - va - va] - t'

\*ot-kry-va-va-t'

TEL-open-A TEL-A TEL-INF

stem-tel-atel-atel-inf

stem - tel - [atel - tel] - inf

stem - tel - [atel - atel] - inf

**Derived strings: 4-SL**

**Derivational strings: 2-SL**

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.

# Russian *-nie* nominalization: a SL account?

The nominalization suffix *-nie* can be added only after the stem was converted to the atelic form.

**atelic**  
<sup>ok</sup>ot-kry-va-t'  
 TEL-open-ATEL-INF

→

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

**telic**  
<sup>ok</sup>na-ot-kry-va-t'  
 TEL-TEL-open-ATEL-INF

→

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

## A SL account?

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

# Russian *-nie* nominalization: a SL account?

The nominalization suffix *-nie* can be added only after the stem was converted to the atelic form.

**atelic**  
<sup>ok</sup>ot-kry-va-t'  
 TEL-open-ATEL-INF

→

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

**telic**  
<sup>ok</sup>na-ot-kry-va-t'  
 TEL-TEL-open-ATEL-INF

→

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

## A SL account?

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

# Russian *-nie* nominalization: a SL account?

The nominalization suffix *-nie* can be added only after the stem was converted to the atelic form.

**atelic**  
<sup>ok</sup>ot-kry-va-t'  
 TEL-open-ATEL-INF

→

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

**telic**  
<sup>ok</sup>na-ot-kry-va-t'  
 TEL-TEL-open-ATEL-INF

→

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

## A SL account?

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

# Russian *-nie* nominalization: a SL account?

The nominalization suffix *-nie* can be added only after the stem was converted to the atelic form.

**atelic**  
<sup>ok</sup>ot-kry-va-t'  
 TEL-open-ATEL-INF

→

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

**telic**  
<sup>ok</sup>na-ot-kry-va-t'  
 TEL-TEL-open-ATEL-INF

→

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

## A SL account?

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

# Russian *-nie* nominalization: a SL account?

The nominalization suffix *-nie* can be added only after the stem was converted to the atelic form.

**atelic**  
<sup>ok</sup>ot-kry-va-t' → <sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-INF TEL-open-ATEL-NMN

**telic**  
<sup>ok</sup>na-ot-kry-va-t' → \*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-INF TEL-TEL-open-ATEL-NMN

## A SL account?

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

# Russian *-nie* nominalization: a SL account?

The nominalization suffix *-nie* can be added only after the stem was converted to the atelic form.

**atelic**  
<sup>ok</sup>ot-kry-va-t' → <sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-INF TEL-open-ATEL-NMN

**telic**  
<sup>ok</sup>na-ot-kry-va-t' → \*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-INF TEL-TEL-open-ATEL-NMN

## A SL account?

<sup>ok</sup>ot-kry-va-nie  
 TEL-open-ATEL-NMN

\*na-ot-kry-va-nie  
 TEL-TEL-open-ATEL-NMN

Russian *-nie* nominalization: derivational account

open-INF → \*open-NMN  
 stem-inf → stem-nmn

TEL-open-INF → \*TEL-open-NMN  
 stem-tel-inf → stem-tel-nmn

TEL-open-ATEL-INF → TEL-open-ATEL-NMN  
 stem-tel-atel-inf → stem-tel-atel-nmn

TEL-TEL-open-ATEL-INF → \*TEL-TEL-open-ATEL-NMN  
 stem-tel-atel-tel-inf → stem-tel-atel-tel-nmn

Russian *-nie* nominalization: derivational account

open-INF → \*open-NMN  
 stem-inf → stem-nmn

TEL-open-INF → \*TEL-open-NMN  
 stem-tel-inf → stem-tel-nmn

TEL-open-A TEL-INF → TEL-open-A TEL-NMN  
 stem-tel-atel-inf → stem-tel-atel-nmn

TEL-TEL-open-A TEL-INF → \*TEL-TEL-open-A TEL-NMN  
 stem-tel-atel-tel-inf → stem-tel-atel-tel-nmn

Russian *-nie* nominalization: derivational account

open-INF → \*open-NMN  
stem-inf → stem-nmn

TEL-open-INF → \*TEL-open-NMN  
stem-tel-inf → stem-tel-nmn

TEL-open-ATEL-INF → TEL-open-ATEL-NMN  
stem-tel-atel-inf → stem-tel-atel-nmn

TEL-TEL-open-ATEL-INF → \*TEL-TEL-open-ATEL-NMN  
stem-tel-atel-tel-inf → stem-tel-atel-tel-nmn

Russian *-nie* nominalization: derivational account

open-INF → \*open-NMN  
 stem-inf → stem-nmn

TEL-open-INF → \*TEL-open-NMN  
 stem-tel-inf → stem-tel-nmn

TEL-open-ATEL-INF → TEL-open-ATEL-NMN  
 stem-tel-atel-inf → stem-tel-atel-nmn

TEL-TEL-open-ATEL-INF → \*TEL-TEL-open-ATEL-NMN  
 stem-tel-atel-tel-inf → stem-tel-atel-tel-nmn

# Russian nominalization: SL derivations

- \*tel-nmn: the telic form cannot be nominalized;

\*na-ot-kry-va-nie  
stem-tel-atel-tel-nmn

- \*stem-nmn: prohibit nominalization of the verbal root.

\*kry-nie  
stem-nmn

# Russian nominalization: SL derivations

- \*tel-nmn: the telic form cannot be nominalized;

\*na-ot-kry-va-nie  
stem-tel-atel-tel-nmn

- \*stem-nmn: prohibit nominalization of the verbal root.

\*kry-nie  
stem-nmn

# Derived vs derivational representations

<sup>ok</sup> **ot-kry-va-nie**

TEL-open-ATEL-NMN

stem-tel-atel-nmn

{ot - kry - va - nie}

{na - ot - kry - va - nie}

**\*na-ot-kry-va-nie**

TEL-open-ATEL-TEL-NMN

stem-tel-atel-tel-nmn

stem - tel - {atel - nmn}

stem - tel - atel - {tel - nmn}

**Derived strings: 5-SL**

**Derivational strings: 2-SL**

The same difference: memorizing the illicit string vs. capturing the generalization.

# Derived vs derivational representations

<sup>ok</sup>ot-kry-va-nie

TEL-open-ATEL-NMN

stem-tel-atel-nmn

ot - kry - va - nie

na - ot - kry - va - nie

Derived strings: 5-SL

\*na-ot-kry-va-nie

TEL-open-ATEL-TEL-NMN

stem-tel-atel-tel-nmn

stem - tel - atel - nmn

stem - tel - atel - tel - nmn

Derivational strings: 2-SL

The same difference: memorizing the illicit string vs. capturing the generalization.

# Derived vs derivational representations

<sup>ok</sup>ot-kry-va-nie

TEL-open-ATEL-NMN

stem-tel-atel-nmn

ot - kry - va - nie

na - ot - kry - va - nie

\*na-ot-kry-va-nie

TEL-open-ATEL-TEL-NMN

stem-tel-atel-tel-nmn

stem - tel - atel - nmn

stem - tel - atel - tel - nmn

**Derived strings: 5-SL**

**Derivational strings: 2-SL**

The same difference: memorizing the illicit string vs. capturing the generalization.

# Derived vs derivational representations

<sup>ok</sup>ot-kry-va-nie

TEL-open-ATEL-NMN

stem-tel-atel-nmn

ot - kry - va - nie

na - ot - kry - va - nie

\*na-ot-kry-va-nie

TEL-open-ATEL-TEL-NMN

stem-tel-atel-tel-nmn

stem - tel - atel - nmn

stem - tel - atel - tel - nmn

**Derived strings: 5-SL**

**Derivational strings: 2-SL**

The same difference: memorizing the illicit string vs. capturing the generalization.

# Derived vs derivational representations

<sup>ok</sup>ot-kry-va-nie

TEL-open-ATEL-NMN

stem-tel-atel-nmn

ot - kry - va - nie

na - ot - kry - va - nie

\*na-ot-kry-va-nie

TEL-open-ATEL-TEL-NMN

stem-tel-atel-tel-nmn

stem - tel - atel - nmn

stem - tel - atel - tel - nmn

**Derived strings: 5-SL**

**Derivational strings: 2-SL**

The same difference: memorizing the illicit string vs. capturing the generalization.

# Derived vs derivational representations

<sup>ok</sup>ot-kry-va-nie

TEL-open-A TEL-NMN

stem-tel-atel-nmn

ot - kry - va - nie

na - ot - kry - va - nie

\*na-ot-kry-va-nie

TEL-open-A TEL-TEL-NMN

stem-tel-atel-tel-nmn

stem - tel - atel - nmn

stem - tel - atel - tel - nmn

**Derived strings: 5-SL**

**Derivational strings: 2-SL**

The same difference: memorizing the illicit string vs. capturing the generalization.

# Conclusion

Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

## From a linguistics perspective:

- derivational representations highlight generalizations;

## From a computational perspective:

- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.

# Conclusion

Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

## From a linguistics perspective:

- derivational representations highlight generalizations;

## From a computational perspective:

- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.

# Conclusion

Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

## From a linguistics perspective:

- derivational representations highlight generalizations;

## From a computational perspective:

- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.

# Conclusion

Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

## From a linguistics perspective:

- derivational representations highlight generalizations;

## From a computational perspective:

- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.

# Future work

## Future work

- extend coverage of empirical phenomena;
- design a learning algorithm;
- explore parallels to subregular syntax (Graf & Heinz 2015).

The more complex the mind, the greater the need for the simplicity of play.

*James T. Kirk*

# Future work

## Future work

- extend coverage of empirical phenomena;
- design a learning algorithm;
- explore parallels to subregular syntax (Graf & Heinz 2015).

The more complex the mind, the greater the need for the  
simplicity of play.

*James T. Kirk*

# References I



Aksënova, Alëna, Thomas Graf and Sedigheh Moradi (2016)  
Morphotactics as Tier-Based Strictly Local Dependencies.  
*In Proceedings of SIGMorPhon 2016.*



Beesley, Kenneth and Lauri Karttunen (2003)  
Finite state morphology.  
Stanford, CA: CSLI Publications.



Kaplan, Ronald M. and Martin Kay (1994)  
Regular Models of Phonological Rule Systems.  
*Computational Linguistics* 20(3), 331 – 378.



Graf, Thomas and J. Heinz (2015)  
*Commonality in Disparity: The Computational View of Syntax and Phonology.*  
Slides of a talk given at GLOW 2015, April 18, Paris, France.



Heinz, Jeffrey and W. Idsardi (2013)  
What Complexity Differences Reveal About Domains in Language.  
*Topics in Cognitive Science*, 5(1):111-131.



McGregor, William B. (2003)  
A fundamental misconception of modern linguistics  
*Acta Linguistica Hafniensia*, 35, 39?64.



McNaughton Robert and S. Papert (1971)  
Counter-Free Automata.  
MIT Press, Cambridge.

# References II



Rogers, J., J. Heinz, G. Bailey, M. Edlefsen, M. Visscher, D. Wellcome, and S. Wibel (2010)

On Languages Piecewise Testable in the Strict Sense

Chapter in LNCS. Springer, 2010.



Shieber, Stuart M. 1985

Evidence against the context-freeness of natural language.

*Linguistics and Philosophy*. 8:333–345.



Vikner, Carl and S. Vikner (2008)

Hierarchical Morphological Structure and Ambiguity

In *L'énonciation dans tous ses états: Mélanges offerts à Henning Nøлке*. ed. Merete Birkelund; Maj-Britt Mosegaard Hansen; Coco Norén. Peter Lang, p. 541-560.

## Non-SL morphotactics: circumfixation

- English **un-...-able** are prefix and suffix that *can* co-occur
- However, two parts of a **circumfix** *cannot* occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- Circumfix **ke-...-an**, “abstract nominalizer”
  - Surrounds the stem:  
tinggi ‘high’ → ke-tinggi-an ‘altitude’
  - ... or multiple stems:  
maha-siswa ‘big-pupil’ → ke-maha-siswa-an ‘student affairs’
  - Parts of this affix cannot occur independently:  
\*ke-tinggi, \*maha-siswa-an
- This pattern is **not SL**: the relations between **ke-** and **-an** are *not local*.

## Non-SL morphotactics: circumfixation

- English **un-...-able** are prefix and suffix that *can* co-occur
- However, two parts of a **circumfix** *cannot* occur independently

### Example (Indonesian circumfixation, Sneddon (1996))

- Circumfix **ke-...-an**, “abstract nominalizer”
  - Surrounds the stem:  
tinggi ‘high’ → ke-tinggi-an ‘altitude’
  - ... or multiple stems:  
maha-siswa ‘big-pupil’ → ke-maha-siswa-an ‘student affairs’
  - Parts of this affix cannot occur independently:  
\*ke-tinggi, \*maha-siswa-an
- This pattern is **not SL**: the relations between ke- and -an are *not local*.

## Non-SL morphotactics: circumfixation

- English **un-...-able** are prefix and suffix that *can* co-occur
- However, two parts of a **circumfix** *cannot* occur independently

### Example (Indonesian circumfixation, Sneddon (1996))

- Circumfix **ke-...-an**, “abstract nominalizer”
  - Surrounds the stem:  
tinggi ‘high’ → ke-tinggi-an ‘altitude’
  - ... or multiple stems:  
maha-siswa ‘big-pupil’ → ke-maha-siswa-an ‘student affairs’
  - Parts of this affix cannot occur independently:  
\*ke-tinggi, \*maha-siswa-an
- This pattern is **not SL**: the relations between **ke-** and **-an** are *not local*.

# TSL morphotactics: circumfixation

**Tier-based strictly local (TSL) grammars** capture long-distance dependencies locally by projecting relevant items on a *tier*.

$$G_{TSL} = \langle \begin{array}{l} T \subseteq \Sigma \quad \# \text{ set of items that are projected on a tier} \\ R \quad \quad \quad \# \text{ set of } k\text{-local strings that are blocked over the tier} \end{array} \rangle$$

## Example (Indonesian circumfixation)

- Elements of the circumfix are projected on a tier.
- $G = \langle \{ke, an\}, \{*an-ke, *ke\_\_, *\_an, *an-an, *ke-ke\} \rangle$
- Intervening stems are ignored, therefore the two parts of the circumfix are *local over the tier*.

# TSL morphotactics: circumfixation

**Tier-based strictly local (TSL) grammars** capture long-distance dependencies locally by projecting relevant items on a *tier*.

$$G_{TSL} = \langle \begin{array}{l} T \subseteq \Sigma \quad \# \text{ set of items that are projected on a tier} \\ R \quad \quad \quad \# \text{ set of } k\text{-local strings that are blocked over the tier} \end{array} \rangle$$

## Example (Indonesian circumfixation)

- Elements of the circumfix are projected on a tier.
- $G = \langle \{ke, an\}, \{*an-ke, *ke\_\times, *_{\times}an, *an-an, *ke-ke\} \rangle$
- Intervening stems are ignored, therefore the two parts of the circumfix are *local over the tier*.

# TSL morphotactics: circumfixation

**Tier-based strictly local (TSL) grammars** capture long-distance dependencies locally by projecting relevant items on a *tier*.

$$G_{TSL} = \langle \begin{array}{l} T \subseteq \Sigma \quad \# \text{ set of items that are projected on a tier} \\ R \quad \quad \quad \# \text{ set of } k\text{-local strings that are blocked over the tier} \end{array} \rangle$$

## Example (Indonesian circumfixation)

- Elements of the circumfix are projected on a tier.
- $G = \langle \{ke, an\}, \{*an-ke, *ke\_\times, *_{\times}an, *an-an, *ke-ke\} \rangle$
- Intervening stems are ignored, therefore the two parts of the circumfix are *local over the tier*.

# TSL morphotactics: circumfixation [cont.]

- $G = \langle \{ke, an\}, \{*an-ke, *ke\text{X}, *Xan, *an-an, *ke-ke\} \rangle$

*ok* maha-siswa:



*ok* ke-maha-siswa-an:



# TSL morphotactics: circumfixation [cont.]

- $G = \langle \{ke, an\}, \{*an-ke, *ke\text{X}, *Xan, *an-an, *ke-ke\} \rangle$

*ok* maha-siswa:



*ok* ke-maha-siswa-an:



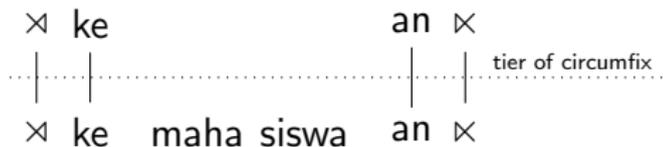
# TSL morphotactics: circumfixation [cont.]

- $G = \langle \{ke, an\}, \{*an-ke, *ke\text{X}, *Xan, *an-an, *ke-ke\} \rangle$

*ok* maha-siswa:



*ok* ke-maha-siswa-an:



# TSL morphotactics: circumfixation [cont.]

- $G = \langle \{ke, an\}, \{*an-ke, *ke\text{X}, *Xan, *an-an, *ke-ke\} \rangle$

\*ke-maha-siswa:



\*maha-siswa-an:



# TSL morphotactics: circumfixation [cont.]

- $G = \langle \{ke, an\}, \{*an-ke, *ke\text{X}, *Xan, *an-an, *ke-ke\} \rangle$

\*ke-maha-siswa:



\*maha-siswa-an:



# TSL morphotactics: circumfixation [cont.]

- $G = \langle \{ke, an\}, \{*an-ke, *ke\text{X}, *Xan, *an-an, *ke-ke\} \rangle$

\*ke-maha-siswa:



\*maha-siswa-an:

