Morphotactics as Tier-Based Strictly Local Dependencies

Alëna Aksënova, Thomas Graf,
and Sedigheh Moradi

Stony Brook University

SIGMORPHON 14
Berlin, Germany
11. August 2016

Our goal

	Phonology	**Morphology**
Received view	regular Kaplan&Kay (1994)	regular Beesley&Karttunen (2003)
Recent research	subregular Heinz (2015)	**?**

Our goal

	Phonology	**Morphology**
Received view	regular Kaplan&Kay (1994)	regular Beesley&Karttunen (2003)
Recent research	subregular Heinz (2015)	**?**

- Show that **morphotactics is subregular**
- More precisely: **Tier-Based Strictly Local**
- **Consequences**
  - parallels to phonology
  - learnable in the limit from positive text
  - explain typological gaps

Outline

Morphotactics

Definition (Morphotactics)

Restrictions on the linear ordering of morphemes.

- Our focus: morphotactics in underlying representations
  (English) ^{OK}STEM-PL *PL-STEM
- $\Rightarrow$ allomorphy (dogs, peaches) is not considered yet

Computational nature of morphotactics

	Phonology	**Morphology**
Received view	regular	regular
Recent research	subregular	**?**

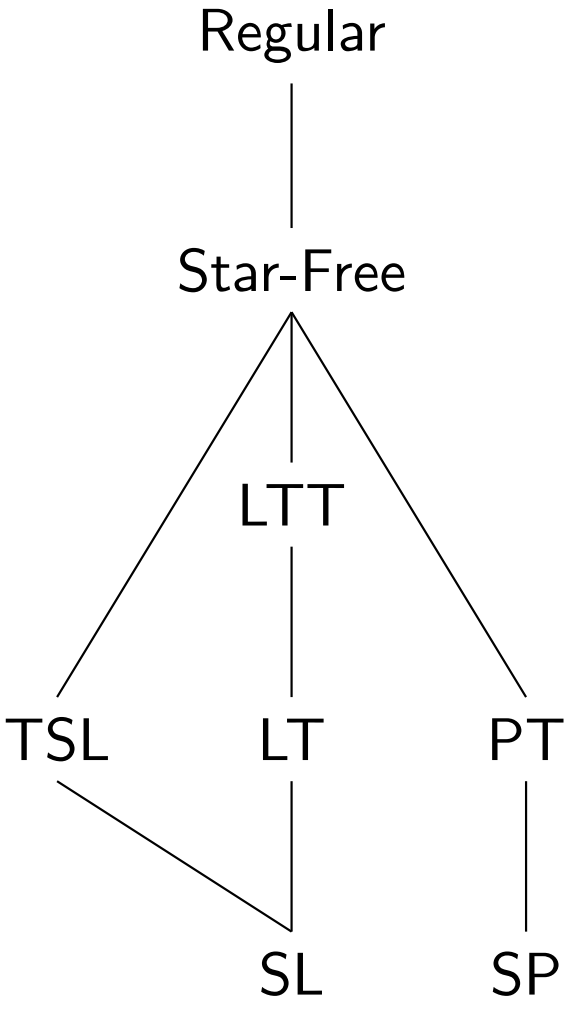
Advantages of (some) subregular languages:

- resolves learnability issues
- describes potential cognitive mechanisms
- uses less powerful generating device

Subregular Phonology and Morphology

not all languages exploit full power of finite-state machinery
$\Rightarrow$ subregular hierarchy

Subregular hierarchy



Subregular Phonology and Morphology

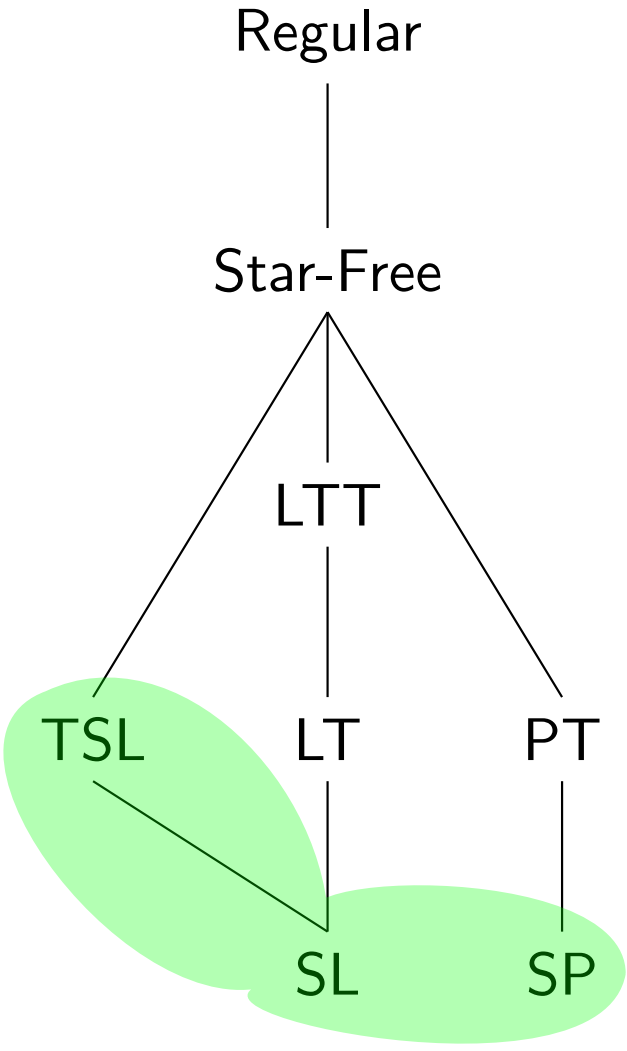
not all languages exploit full power of finite-state machinery
$\Rightarrow$ subregular hierarchy

Strong Subregular Hypothesis

All **phonological dependencies** are

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

Subregular hierarchy



Subregular Phonology and Morphology

not all languages exploit full power of finite-state machinery
$\Rightarrow$ subregular hierarchy

Strong Subregular Hypothesis

All **phonological dependencies** are

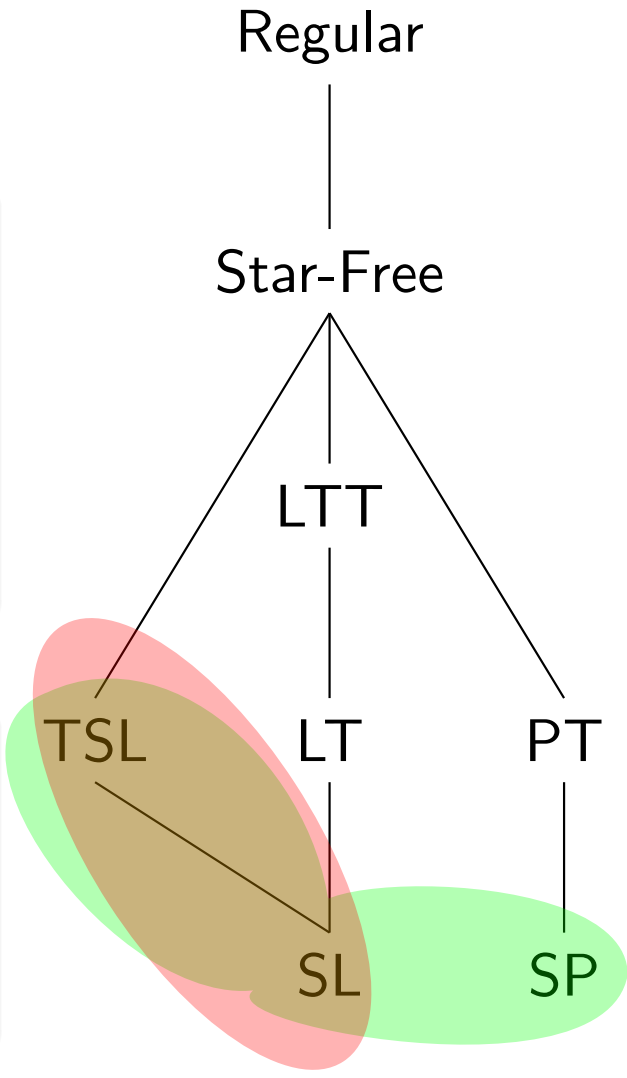
- strictly local (SL)
- tier-based strictly local (TSL)
- strictly piecewise (SP)

Subregular Morphotactics

All **morphotactic** dependencies are

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

Subregular hierarchy



Strictly Local languages

- SL and TSL are generated by ***k*-gram models**.
- A k-gram model is a finite set of blocked k-grams.

Strictly Local languages

- SL and TSL are generated by ***k*-gram models**.
- A k-gram model is a finite set of blocked k-grams.

Example (Strictly Local Grammar for $(ab)^*a$)

$\Sigma = \{\text{a}, \text{b}\}$
Grammar $= \{\rtimes\text{b}, \text{bb}, \text{aa}, \text{b}\ltimes, \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes aba\ltimes$, $\rtimes ababa\ltimes$, etc.
Rejected strings: $\rtimes ab\ltimes$, $\rtimes ba\ltimes$, $\rtimes abba\ltimes$, etc.

Strictly Local languages

- SL and TSL are generated by ***k*-gram models**.
- A k-gram model is a finite set of blocked k-grams.

Example (Strictly Local Grammar for $(ab)^*a$)

$\Sigma = \{\text{a, b}\}$
Grammar $= \{\rtimes\text{b, bb, aa, b}\ltimes, \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes aba\ltimes$, $\rtimes ababa\ltimes$, etc.
Rejected strings: $\rtimes ab\ltimes$, $\rtimes ba\ltimes$, $\rtimes abba\ltimes$, etc.

Strictly Local languages

- SL and TSL are generated by ***k*-gram models**.
- A k-gram model is a finite set of blocked k-grams.

Example (Strictly Local Grammar for $(ab)^*a$)

$\Sigma = \{\text{a}, \text{b}\}$
Grammar $= \{\rtimes\text{b}, \text{bb}, \text{aa}, \text{b}\ltimes, \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes aba\ltimes$, $\rtimes ababa\ltimes$, etc.
Rejected strings: $\rtimes ab\ltimes$, $\rtimes ba\ltimes$, $\rtimes abba\ltimes$, etc.

Definition

Strictly k-Local (k-SL) grammar consists of a set of blocked k-grams over an alphabet Σ.

Tier-Based Strictly Local languages

Example (Tier-Based Strictly Local Grammar for $c^*(ac^*bc^*)^*ac^*$)

$\Sigma = \{a, b, c\}$

Grammar:

$G(a,b\_tier) = \{\rtimes b, bb, aa, b\ltimes, \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes accba\ltimes$, $\rtimes cacbaccccba\ltimes$, etc.

Tier-Based Strictly Local languages

Example (Tier-Based Strictly Local Grammar for $c^*(ac^*bc^*)^*ac^*$)

$\Sigma = \{a, b, c\}$

Grammar:

G(a,b_tier) $= \{\rtimes b, bb, aa, b\ltimes, \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes accba\ltimes$, $\rtimes cacbaccccba\ltimes$, etc.

a,b_tier: $\rtimes a\ltimes$ $\rtimes aba\ltimes$ $\rtimes ababa\ltimes$

Tier-Based Strictly Local languages

Example (Tier-Based Strictly Local Grammar for $c^*(ac^*bc^*)^*ac^*$)

$\Sigma = \{a, b, c\}$

Grammar:

$G(a,b\_tier) = \{\rtimes b,\ bb,\ aa,\ b\ltimes,\ \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes accba\ltimes$, $\rtimes cacbaccccba\ltimes$, etc.

a,b_tier: $\rtimes a\ltimes$ $\rtimes aba\ltimes$ $\rtimes ababa\ltimes$

Rejected strings: $\rtimes accccaba\ltimes$, $\rtimes abcccacccbc\ltimes$, etc.

Tier-Based Strictly Local languages

Example (Tier-Based Strictly Local Grammar for $c^*(ac^*bc^*)^*ac^*$)

$\Sigma = \{a, b, c\}$

Grammar:

G(a,b_tier) $= \{\rtimes b,\ bb,\ aa,\ b\ltimes,\ \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes accba\ltimes$, $\rtimes cacbaccccba\ltimes$, etc.

a,b_tier: $\rtimes a\ltimes$ $\rtimes aba\ltimes$ $\rtimes ababa\ltimes$

Rejected strings: $\rtimes accccaba\ltimes$, $\rtimes abcccacccbc\ltimes$, etc.

a,b_tier: $\rtimes aaba\ltimes$ $\rtimes abab\ltimes$

Tier-Based Strictly Local languages

Example (Tier-Based Strictly Local Grammar for $c^*(ac^*bc^*)^*ac^*$)

$\Sigma = \{a, b, c\}$

Grammar:

G(a,b_tier) $= \{\rtimes b,\ bb,\ aa,\ b\ltimes,\ \rtimes\ltimes\}$

Accepted strings: $\rtimes a\ltimes$, $\rtimes accba\ltimes$, $\rtimes cacbaccccba\ltimes$, etc.

a,b_tier: $\rtimes a\ltimes$ $\rtimes aba\ltimes$ $\rtimes ababa\ltimes$

Rejected strings: $\rtimes accccaba\ltimes$, $\rtimes abcccacccbc\ltimes$, etc.

a,b_tier: $\rtimes aaba\ltimes$ $\rtimes abab\ltimes$

Definition

A Tier-Based Strictly k-Local grammar is a k-SL grammar that operates over a *tier*, a specific substructure of the string.

Learnability

Learning of SL and TSL

- learning $\equiv$ memorizing finite number of k-grams $+$ tier induction
- learnable in the limit from positive text

Jardine & Heinz (2016)

Mappings we use

General assumption: we assume stem not to be bound in length:

- There is no limit on the length of the stem in languages.
- The stem can be result of the compounding.
  whiteboard, whiteboard marker, whiteboard marker cleaning fluid, whiteboard marker cleaning fluid purchase receipt
- Mapping of the stem to a single symbol will result in insensibility to compounds.

Mappings we use

General assumption: we assume stem not to be bound in length:

- There is no limit on the length of the stem in languages.
- The stem can be result of the compounding.
  whiteboard, whiteboard marker, whiteboard marker cleaning fluid, whiteboard marker cleaning fluid purchase receipt
- Mapping of the stem to a single symbol will result in insensibility to compounds.

- Affixes: affix-to-symbol mapping
- Stems: symbol-to-symbol mapping

Strictly Local Morphology: affixation

Example (prefix ‘za-’, Russian)

- exat’
  ‘go, drive’
  xxxx
- zaexat’
  ‘call on the way’
  axxxx

Bigram $^*xa$ ensures that ‘za’ is a prefix.

Strictly Local Morphology: affixation

Example (prefix 'za-', Russian)

- exat'
  'go, drive'
  xxxx
- zaexat'
  'call on the way'
  axxxx

Bigram $^*xa$ ensures that 'za' is a prefix.

Example (suffix '-s', English)

- dog
  xxx
- dogs
  xxxb

Bigram $^*bx$ ensures that 's' is a suffix.

Strictly Local Morphology: affixation [cont.]

Example (affixation, English)

- lock
  xxxx
- unlockable
  axxxxb
- blacklist
  xxxxxxxxx
- unblacklistable
  axxxxxxxxxb

Strictly Local Morphology: affixation [cont.]

Example (affixation, English)

- lock
  xxxx
- unlockable
  axxxxb
- blacklist
  xxxxxxxxx
- unblacklistable
  axxxxxxxxxb

SLG $= \{\rtimes$b, ba, bx, xa, a$\ltimes\}$

This grammar necessarily generates the following forms of English, too: $\rtimes$axxxxx$\ltimes$ and $\rtimes$xxxxxb$\ltimes$.

Strictly Local Morphology: affixation [cont.]

Example (affixation, English)

- lock
  xxxx
- unlockable
  axxxxb
- blacklist
  xxxxxxxxx
- unblacklistable
  axxxxxxxxxb

SLG $= \{\rtimes$b, ba, bx, xa, a$\ltimes\}$

This grammar necessarily generates the following forms of English, too: $\rtimes$axxxxx$\ltimes$ and $\rtimes$xxxxxb$\ltimes$.

Indeed, this prediction is correct:

Example (affixation, English)

- unleash
  axxxxx
- breakable
  xxxxxb

SL is not enough: Indonesian circumfixation

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

Consider the following example from Indonesian:

Example (circumfix 'ke-an', Indonesian)

- tinggi
  'high'
  xxxxxx
- ketinggian
  'altitude'
  axxxxxxb
- *axxxxxx

- mahasiswa
  'big pupil (student)'
  xxxxxxxxx
- kemahasiswaan
  'student affairs'
  axxxxxxxxxb
- *xxxxxxxb

SL is not enough: Indonesian circumfixation [cont.]

Example (circumfix 'ke-an', Indonesian)

- tinggi
  xxxxxx
- ketinggian
  axxxxxxb
- *axxxxxx

- mahasiswa
  xxxxxxxxx
- kemahasiswaan
  axxxxxxxxxb
- *xxxxxxxb

SLG = {⋊b, ba, bx, xa, a⋉}
String language = ⋊*xxxx*⋉, ⋊*axxxxxb*⋉, ⋊*axxxx*⋉, ⋊*xxb*⋉...

SL is not enough: Indonesian circumfixation [cont.]

Example (circumfix 'ke-an', Indonesian)

- tinggi
  xxxxxx
- ketinggian
  axxxxxxb
- *axxxxxx

- mahasiswa
  xxxxxxxxx
- kemahasiswaan
  axxxxxxxxxb
- *xxxxxxxb

SLG = {⋊b, ba, bx, xa, a⋉}
String language = ⋊*xxxx*⋉, ⋊*axxxxxb*⋉, ⋊*axxxx*⋉, ⋊*xxb*⋉...

Problem:

- SL languages can only capture local dependencies
- Circumfixes introduce non-local ones

Morphotactics is TSL

Example (circumfix ‘ke-an’, Indonesian)

- tinggi
  xxxxxx
- ketinggian
  axxxxxxb
- *axxxxxx
- mahasiswa
  xxxxxxxxx
- kemahasiswaan
  axxxxxxxxxb
- *xxxxxxxb

TSLG(circumfix_tier) $= \{\rtimes$b, ba, a$\ltimes\}$

Morphotactics is TSL

Example (circumfix ‘ke-an’, Indonesian)

- tinggi
  xxxxxx
- ketinggian
  axxxxxxb
- *axxxxxx
- mahasiswa
  xxxxxxxxx
- kemahasiswaan
  axxxxxxxxxb
- *xxxxxxxb

TSLG(circumfix_tier) $= \{\rtimes b,\ ba,\ a \ltimes\}$

Licit strings:

- $\rtimes xxxxxx \ltimes$
  $\rtimes \ltimes$
- $\rtimes axxxxxxb \ltimes$
  $\rtimes ab \ltimes$

Illicit strings:

- $\rtimes axxxxx \ltimes$
  $\rtimes a \ltimes$
- $\rtimes bxxxxa \ltimes$
  $\rtimes ba \ltimes$

Morphotactics is TSL

Example (circumfix ‘ka-an’, Ilocano)

In Ilocano, it is impossible to do embedded circumfixation:

- bigát
  ‘morning’
  xxxxx
- kabigátan
  ‘the next morning’
  axxxxxb
- *aaxxxxxbb

Morphotactics is TSL [cont.]

Example (circumfix ‘ka-an’, Ilocano)

- bigát
  xxxxx
- kabigátan
  axxxxxb
- *aaxxxxxbb

$\text{TSLG}(\text{circumfix\_tier}) = \{\rtimes\text{b}, \text{ba}, \text{a}\ltimes, \text{aa}, \text{bb}\}$

Morphotactics is TSL [cont.]

Example (circumfix ‘ka-an’, Ilocano)

- bigát
  xxxxx
- kabigátan
  axxxxxb
- *aaxxxxxbb

TSLG(circumfix_tier) = $\{\rtimes b,\ ba,\ a\ltimes,\ aa,\ bb\}$

Licit strings:

- $\rtimes xxxxxx\ltimes$
  $\rtimes\ltimes$
- $\rtimes axxxxxxb\ltimes$
  $\rtimes ab\ltimes$

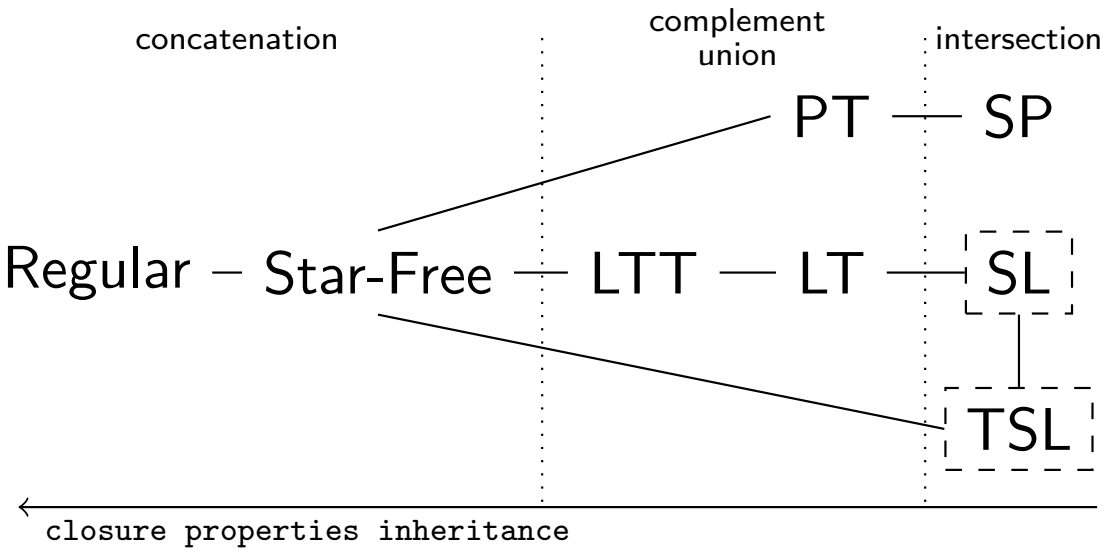
Illicit strings:

- $\rtimes aaxxxxxbb\ltimes$
  $\rtimes aabb\ltimes$
- $\rtimes bxxxxa\ltimes$
  $\rtimes ba\ltimes$

Interim Summary

- SL enforces local dependencies
- TSL enforces local dependencies on the determined tier
- Most of morphotactics is SL, some of it is TSL
- Learning of TSL languages is possible from positive data only
- Can morphotactics be more than TSL?

Can morphotactics be more than TSL?



Can morphotactics be more than TSL?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English.

Can morphotactics be more than TSL?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗

Can morphotactics be more than TSL?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗
- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian.

Can morphotactics be more than TSL?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗
- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian. ✗

Can morphotactics be more than TSL?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗
- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian. ✗
- **Closure under relative complement**: Hsilgne contains all words that are ill-formed in English.

Can morphotactics be more than TSL?

- **Closure under concatenation**: Frenglish contains only words whose first part is a word of French and the second a word of English. ✗
- **Closure under union**: If a Mandaresian word violates rules of Mandarin Chinese, it must obey the rules of Indonesian. ✗
- **Closure under relative complement**: Hsilgne contains all words that are ill-formed in English. ✗

Can morphotactics be more than TSL?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian.

Can morphotactics be more than TSL?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian. ✓
  (spoken in Northern Norway, 18th-19th centuries)

Can morphotactics be more than TSL?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian. ✓
  (spoken in Northern Norway, 18th-19th centuries)

Example (Closure under intersection)

- A language allows complex nuclei and blocks codas (Supyire)
- A language forbids complex nuclei and allows codas (Russian)

Can morphotactics be more than TSL?

- **Closure under intersection**: Russenorsk is created by combination of elements of Russian and Norwegian. ✓
  (spoken in Northern Norway, 18th-19th centuries)

Example (Closure under intersection)

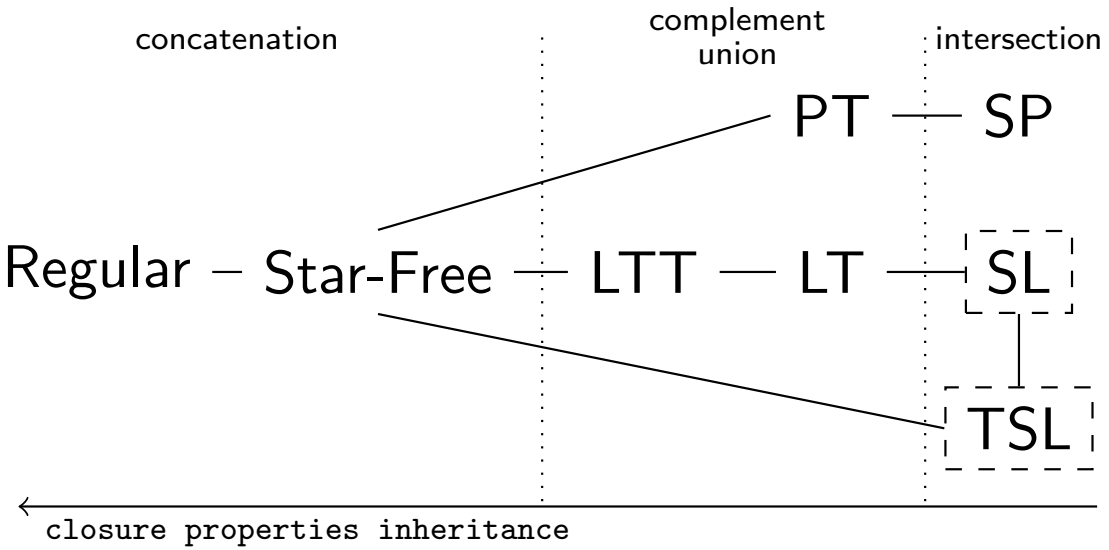
- A language allows complex nuclei and blocks codas (Supyire)
- A language forbids complex nuclei and allows codas (Russian)
- Then there will be a language that blocks complex nuclei and codas (Hawaiian, Senufo)

Can morphotactics be more than TSL?

- ✗ Closure under concatenation
- ✗ Closure under union
- ✗ Closure under relative complement
- ✓ Closure under intersection

Can morphotactics be more than TSL?

- ✗ Closure under concatenation
- ✗ Closure under union
- ✗ Closure under relative complement
- ✓ Closure under intersection



Typological gaps

Basic Logic of Argument

- All attested morphotactic patterns must be TSL.
- So if pattern A is TSL, and pattern B is TSL, but their combination A+B is not, we get a typological gap.

Some predicted gaps:

- No embedded circumfixation;
- No cases when amount of affixes A depends on the amount of affixes B;
- In general, no $a^n b^n$ pattern and its derivatives.

Typological gap I: Impossible compounding

Russian pattern – (stem-o)*-stem

Example (compounding, Russian)

- vodovoz
  ‘water carrier’
  xxxoxxx
- vodovozovoz
  ‘carrier of water carriers’
  xxxoxxxoxxx

Typological gap I: Impossible compounding

Russian pattern – (stem-o)*-stem

Example (compounding, Russian)

- vodovoz
  'water carrier'
  xxxoxxx
- vodovozovoz
  'carrier of water carriers'
  xxxoxxxoxxx

Turkish pattern – stem-(stem^{+}-o)

Example (compounding, Turkish)

- bahçe kapı-sı
  'garden gate'
  xxxxxxxxxo
- türk kahve-sı
  'Turkish coffee'
  xxxxxxxxo
- türk bahçe kapı-sı
  'Turkish garden gate'
  xxxxxxxxxxxxo
- *türk bahçe kapı-sı-sı
  *xxxxxxxxxxxxoo

Typological gap I: Impossible compounding

Russian pattern – (stem-o)*-stem
Turkish pattern – stem-(stem^{+}-o)

Turkussian pattern: amount of compound markers is equal to the amount of added stems, stem-(stem^{n}-o^{n})

Typological gap I: Impossible compounding

Russian pattern – (stem-o)*-stem
Turkish pattern – stem-(stem^{+}-o)

Turkussian pattern: amount of compound markers is equal to the amount of added stems, stem-(stem^{n}-o^{n})

- This pattern is not regular because it has infinite number of "good continuations". (*Myhill-Nerode theorem*)
- It appears to be non-existent.

Typological gap II: Recurrent affixation

Sometimes languages allow some affixes to be iterated: a*-stem.

Consider example of such pattern in German:

Example (prefix 'über', German)

- morgen
  'tomorrow'
  xxxxxx
- übermorgen
  'the day after tomorrow'
  axxxxxx
- überübermorgen
  'the day after the day after tomorrow'
  aaxxxxxx

Typological gap II: Recurrent affixation

German pattern: a*-stem.

The same meaning can be expressed in another language differently, consider Ilocano (Austronesian) temporal circumfix *ka-...-an* 'next'.

Example (circumfix 'ka-an', Ilocano)

- bigát
  'morning'
  xxxxx
- kabigátan
  'the next morning'
  axxxxxb

Typological gap II: Recurrent affixation

German pattern: a*-stem.

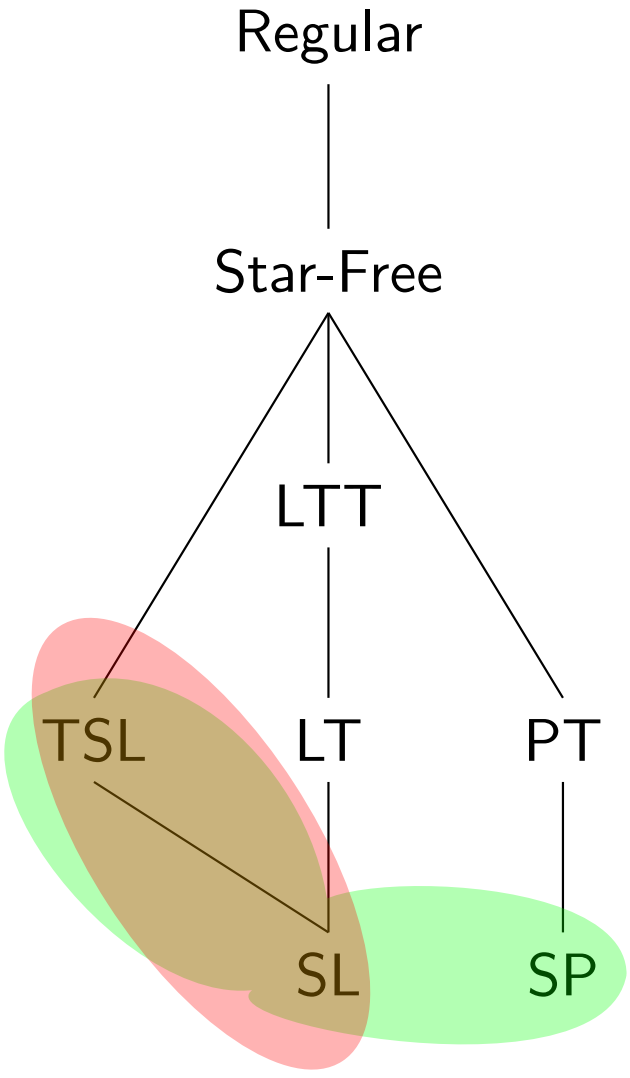
The same meaning can be expressed in another language differently, consider Ilocano (Austronesian) temporal circumfix *ka-...-an* 'next'.

Example (circumfix 'ka-an', Ilocano)

- bigát
  'morning'
  xxxxx
- kabigátan
  'the next morning'
  axxxxxb

However, word kakabigátanan doesn't appear to be possible word in Ilocano: a^n-stem-b^n pattern is not regular.

Conclusion



- Morphotactics is at most Tier-Based Strictly Local
- Positive data is enough for morphological learning
- Set of typological gaps can be explained due to the subregular nature of morphology
- Same formal tools can be used for **morphology** and **phonology**

Future work

- Try to find SP patterns in morphotactics
- Look at more typologically diverse languages
- Extend to mappings from underlying to surface forms
- Work with representations of internal structure
- The elephant in the room: reduplication

Thank you!

References I

Beesley, Kenneth R. and Lauri Karttunen (2003)

Finite State Morphology.
CSLI Publications.

Chandlee, Jane (2014)

Strictly Local Phonological Processes.
PhD Thesis, University of Delaware

Chandlee, Jane, Rémi Eyraud and Jeffrey Heinz (2014)

Learning Strictly Local Subsequential Functions.
Transactions of the Association for Computational Linguistics 2, 491 – 503.

Galvez Rubino, Carl R. (1998)

Ilocano: Ilocano-English, English-Ilocano: Dictionary and Phrasebook.
Hippocrene Books Inc., U.S.

Heinz, Jeffrey (2015)

The Computational Nature of Phonological Generalizations.
Ms., University of Delaware

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

Tier-Based Strictly Local Constraints in Phonology.
Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 58 – 64.

Jardine, Adam (2015)

Computationally, Tone is Different.
Ms., University of Delaware

References II

Jardine, Adam and Jeffrey Heinz (2016)

Learning Tier-based Strictly 2-Local Languages.
Transactions of the Association for Computational Linguistics 4, 87 – 98.

Jurafsky, Daniel and James H. Martin. (2009)

Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition.
Upper Saddle River, N.J. : Pearson Prentice Hall.

Kaplan, Ronald M. and Martin Kay (1994)

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

Mahdi, Waruno (2012)

Distinguishing Cognate Homonyms in Indonesian.
Oceanic Linguistics 51(2), 402 – 449.

Rogers, James and Geoffrey Pullum (2007)

Aural Pattern Recognition Experiments and the Subregular Hierarchy.
Mathematics of Language 10, 1 – 16.

Sneddon, James Neil (1996)

Indonesian Comprehensive Grammar.
Routledge, London and New York.

Stump, Greg (2016)

Rule composition in an adequate theory of morphotactics.
Manuscript.