Insertion without competition: a case for Free Choice

Pavel Caha[1] Karen De Clercq[2] Guido Vanden Wyngaerd[3]

[1]Masarykova Univerzita

[2]FWO/U Ghent (Ghent)

[3]KU Leuven (Brussels)

Olomouc Linguistics Colloquium
7-9 June 2018

Introduction

- (1) presents comparative and superlative morphology in Czech
- this talk: comparative morphology

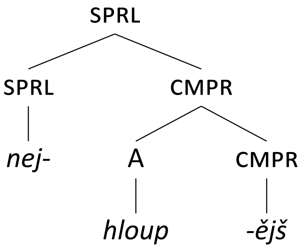
(1)

	POS	CMPR	SPRL	
	bujar-ý	bujař-ejš-í	nej-bujař-ejš-í	‘merry’
	červen-ý	červen-ějš-í	nej-červen-ějš-í	‘red’
	hloup-ý	hloup-ějš-í	nej-hloup-ějš-í	‘stupid’
	moudr-ý	moudř-ejš-í	nej-moudř-ejš-í	‘wise’

í/ý = adjectival agreement: Case, number, gender

Introduction

(2)



Introduction

Claims

- the comparative consists of two heads (C1 and C2) instead of one CMPR head
- adjectives come in various sizes, spelling out different layers of functional structure
- the existence of roots of various sizes requires us to rethink the way competition between candidates for spellout works
- the Elsewhere Principle can be dispensed with
- a Faithfulness Restriction on Cyclic Override is needed

Three allomorphs of the Czech comparative

(3)	I	*ějš-í*
	II	*š-í*
	III	*-í*

Three allomorphs of the Czech comparative

- allomorph I: ějš

(4)

POS	CMPR	
mil-ý	mil-ejš-í	‘nice/kind’
kulat-ý	kulat-ějš-í	‘round’
hloup-ý	hloup-ějš-í	‘stupid’
bujar-ý	bujař-ejš-í	‘wild’
benign-í	benign-ějš-í	‘benign’

Three allomorphs of the Czech comparative

- allomorph II: š

(5)

POS	CMPR	
star-ý	star-š-í	‘old’
tvrd-ý	tvrd-š-í	‘hard’
tich-ý	tiš-š-í	‘silent’
drah-ý	draž-š-í	‘expensive’
bohat-ý	bohat-š-í	‘rich’

Three allomorphs of the Czech comparative

- allomorph IIIa: ø

(6)

POS	CMPR	
leh-k-ý	leh-č-ø-í	‘light/easy’
hez-k-ý	hez-č-ø-í	‘pretty’
měk-k-ý	měk-č-ø-í	‘soft’
ten-k-ý	ten-č-ø-í	‘thin’
vlh-k-ý	vlh-č-ø-í	‘wet’

- k palatalises to č under the influence of the soft declension marker í
- palatalisation of k to č under the influence of í is also seen elsewhere in Czech

(7) vlk → vlk-í → vlč-í
wolf.N → wolf.A

Three allomorphs of the Czech comparative

- there is a conceivable alternative analysis, which assumes an underlying comparative marker *š*

(8)

POS	CMPR	
leh-k-ý	leh-č-š-í	'light/easy'
hez-k-ý	hez-č-š-í	'pretty'
měk-k-ý	měk-č-š-í	'soft'
ten-k-ý	ten-č-š-í	'thin'
vlh-k-ý	vlh-č-š-í	'wet'

- palatalisation of k to č under the influence of š is also seen elsewhere in Czech
- š gets deleted after *č*

(9) nor → nor-ští
Norwegian.N → Norwegian.A

(10) turek → turek-ští → tureč-ští → tureč-tí
Turk.N → Turk-ish

Three allomorphs of the Czech comparative

- allomorph IIIb: ø

(11)

POS	St. Czech CMPR	N-E. Bohemian CMPR	
ostr-ý	ostř-ejš-í	ostř-ø-í	‘sharp’
mokr-ý	mokř-ejš-í	mokř-ø-í	‘wet’

- r palatalises to ř under the influence of the soft declension marker í

(12) vydr-a → vydr-í → vydř-í
otter.N → otter.A

Three allomorphs of the Czech comparative

- an alternative analysis would explain palatalised ř as a consequence of the presence of an underlying comparative marker š

(13)

POS	St. Czech CMPR	N-E. Bohemian CMPR	
ostr-ý	ostř-ejš-í	ostř-š-í	‘sharp’
mokr-ý	mokř-ejš-í	mokř-š-í	‘wet’

- in this case, this alternative seems unlikely
- rš clusters are preserved as is

(14) bratr → bratr-ští → *bratř-tí
brother.N → brother-ly

Three allomorphs of the Czech comparative

(15)	I	*ějš-í*
	II	*š-í*
	III	*-í*

- the allomorphs II and III correspond to various degrees of the reduction of I going from left to right
- we present a theory that explains how such a reduction works

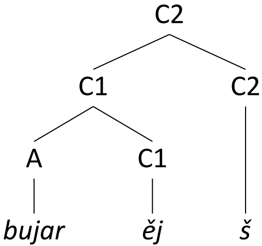
Three allomorphs of the Czech comparative

- the suffix *-ějš* splits into *-ěj* and *-š*
- there are two comparative heads, C1 and C2
- *-ěj-* spells out C1, and *-š-* spells out C2

Three allomorphs of the Czech comparative

- allomorph I: ějš

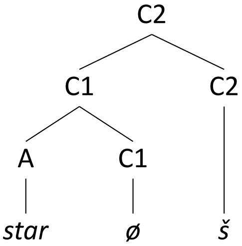
(16) The *-ějš*-comparative



Three allomorphs of the Czech comparative

- allomorph II: š

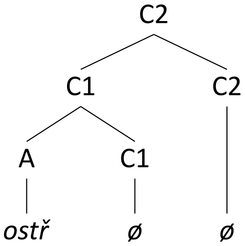
(17) The -*š*-comparative



Three allomorphs of the Czech comparative

- allomorph IIIb: ø

(18) The *ø*-comparative



Three allomorphs of the Czech comparative

- independent evidence for the decomposition of *-ějš-*
- comparative adverbs systematically miss the *-š*-part of the comparative adjective

(19)

CMPR ADJ	CMPR ADV	
rychl-ej-š-í	rychl-ej-i	'faster'
červen-ěj-š-í	červen-ěj-i	'redder'
hloup-ěj-š-í	hloup-ěj-i	'sillier'
bujař-ej-š-í	bujař-ej-i	'merrier'

The distribution of zeroes depending on the root type

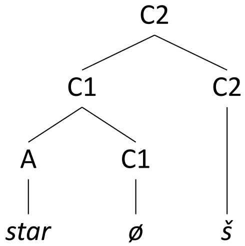
(20)

		√	C1	C2	
a.	ATTESTED:	$\sqrt{\textit{Type 1}}$	/α/	/β/	(*bujar-ěj-š-í*)
b.	ATTESTED:	$\sqrt{\textit{Type 2}}$	ø	/α/	(*star-ø-š-í*)
c.	ATTESTED:	$\sqrt{\textit{Type 3}}$	ø	ø	(*ostř-ø-ø-í*)
d.	NOT ATTESTED:	$\sqrt{\textit{Type 4}}$	/α/	ø	

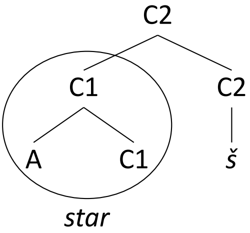
The distribution of zeroes depending on the root type

- ‘zero exponents’ arise as a consequence of phrasal spellout
- a single lexical item may realise multiple positions in the syntactic/morphological structure

(21) The old proposal

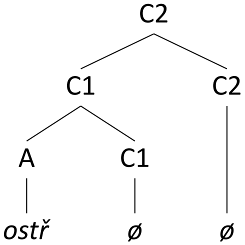


(22) The new proposal

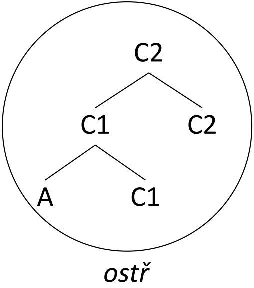


The distribution of zeroes depending on the root type

(23) The old proposal



(24) The new proposal



The distribution of zeroes depending on the root type

- How is it that adjectives like *star-* 'old' (C1P) and *ostr-* 'sharp' (C2P) can appear in the positive degree?
- these adjectives are larger than the syntactic structure of the positive degree

(25) *The Superset Principle* (SP) (Starke 2009)
A lexically stored L-tree can spell out a syntactic S-tree iff the L-tree contains the S-tree as a subtree.

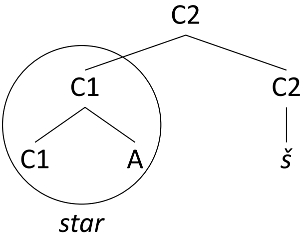
- both C2P and C1P include the positive degree (A) as a subtree, hence these roots can spell out the positive degree

Suppletion

- decomposed CMPR allows an elegant account of root suppletion
  - *dobr-ý* ‘good’
  - *lep-š-í* ‘bett-er’
- recall roots of the *star-* ‘old’ class:

(26)

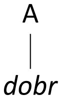
(27)



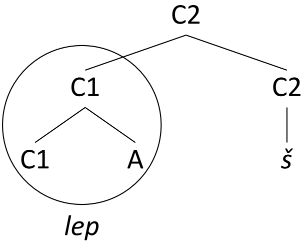
Suppletion

- *dobr-* ‘good’ spells out A
- *lep-* ‘bett-’ spells out C1+A, and *-š* spells out C2

(28)



(29)

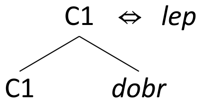


Suppletion

- in the lexicon, the entry for *lep-* ‘bett-’ contains a pointer to *dobr-* ‘good’

(30) A ⇔ *dobr*

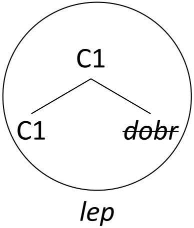
(31)



Suppletion

- the syntax merges QP, and spells it out as *dobr*- 'good'
- if the syntax then goes on to merge C1P, *lep*- 'bett-' can be inserted if at the previous cycle *dobr*- was inserted
- *lep*- overrides the earlier spellout *dobr*- (Cyclic Override)

(32)



Decomposing A

- allomorph IIIa occurs with adjectives ending in *-k*
- *-k* is a derivational suffix

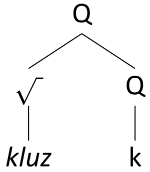
(33)

POS	CMPR	GLOSS OF A	BASE	GLOSS
břit-k-ý	břit-č-í	‘sharp’	břit	‘edge’
hoř-k-ý	hoř-č-í	‘bitter’	hoř-e	‘sorrow’
kluz-k-ý	kluz-č-í	‘slippery’	s-kluz	‘a slide’
sliz-k-ý	sliz-č-í	‘slimy’	sliz	‘slime’
vlh-k-ý	vlh-č-í	‘wet’	vláh-a	‘dew’
ten-k-ý	ten-č-í	‘thin’		

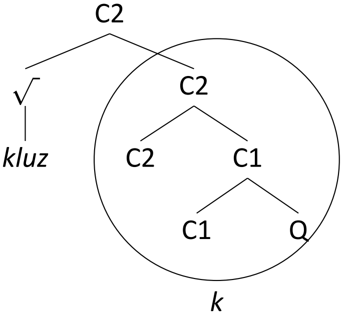
Decomposing A

- the head A has internal structure, and is composed of
  - a root feature $\sqrt{\ }$
  - a gradability feature Q

(34)



(35)



The spellout algorithm

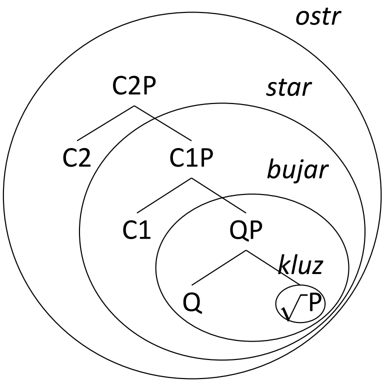
- roots come in varying sizes
- the allomorphy of the affixes is a function of the size of the root

(36)

<table>
<tr><th>√</th><th>Q</th><th>C1</th><th>C2</th><th></th></tr>
<tr><td>kluz</td><td colspan="3">k</td><td>‘slippery’</td></tr>
<tr><td colspan="2">bujar</td><td>ěj</td><td>š</td><td>‘merry’</td></tr>
<tr><td colspan="3">star</td><td>š</td><td>‘old’</td></tr>
<tr><td colspan="4">ostř</td><td>‘sharp’ (NE Bohemian)</td></tr>
</table>

The spellout algorithm

(37)



The spellout algorithm

(38)
a. $[_{\sqrt{}P}\ \sqrt{}\ ]$ ⇔ /kluz/
b. $[_{QP}\ Q\ [_{\sqrt{}P}\ \sqrt{}\ ]]$ ⇔ /bujar/
c. $[_{C1P}\ C1\ [_{QP}\ Q\ [_{\sqrt{}P}\ \sqrt{}\ ]]]$ ⇔ /star/
d. $[_{C2P}\ C2\ [_{C1P}\ C1\ [_{QP}\ Q\ [_{\sqrt{}P}\ \sqrt{}\ ]]]]$ ⇔ /ostř/

(39)
a. $[_{C1P}\ C1\ ]$ ⇔ /ěj/
b. $[_{C2P}\ C2\ ]$ ⇔ /š/
c. $[_{C2P}\ C2\ [_{C1P}\ C1\ [_{QP}\ Q\ ]]]$ ⇔ /k/

The spellout algorithm

Starke (2018)

(40) Merge F and

a. Spell out FP

b. If (a) fails, attempt any of the rescue strategies below (in the order given), and retry (a), until spellout is successful

(i) move the spec of the complement of F

(ii) move the complement of F

(iii) start a new derivation by merging F with the last successfully spelled out feature, i.e. F^{-1}

The spellout algorithm

(41) Construct $\sqrt{}$P and spell-out

a. [$\sqrt{}$P]

b. [C1 [Q [$\sqrt{}$P]]] ⇔ star

(42) Merge Q and spell out

a. [Q [$\sqrt{}$P]]

b. [C1 [Q [$\sqrt{}$P]]] ⇔ star

(43) Merge C1 and spell out

a. [C1 [Q [$\sqrt{}$P]]]

b. [C1 [Q [$\sqrt{}$P]]] ⇔ star

(44) Merge C2 and spell out

a. [C2 [C1 [Q [$\sqrt{}$P]]]]

b. [C2] ⇔-š

c. [C1 [Q [$\sqrt{}$P]]] [C2 ...] = *star-š*

The spellout algorithm

(45) Construct $\sqrt{\ }$P and spell-out

a. [$\sqrt{\ }$P]

b. [$\sqrt{\ }$P] ⇔ kluz

(46) Merge Q and spell out

a. [Q [$\sqrt{\ }$P]]

b. [C2 [C1 [Q]]] ⇔ -k

c. [$\sqrt{\ }$P] [Q ...] = *kluz -k*

(47) Merge C1 and spell out

a. [C1 [[$\sqrt{\ }$P] [Q ...]]]

b. [C2 [C1 [Q]]] ⇔ -k

c. [$\sqrt{\ }$P] [C1 ... [Q ...]] = *kluz-k*

Competition between roots

- Two problems relating to the competition between roots of different sizes:
  - Limited Free Choice: in the positive degree, roots which spell out C1P (e.g. *star* ‘old’) will lose against QP roots (e.g. *bujar* ‘merry’)
  - Faithfulness: ‘small’ roots will be overridden by larger ones, which are unfaithful to the initial choice

Limited Free Choice

- suppose the syntax merges QP, and consults the lexicon
- three different root types are candidates for insertion by SP
  - *bujar*-type (QP)
  - *star*-type (C1P)
  - *ostr*-type (C2P)
- the *Elsewhere Principle* determines the outcome

(48) *Elsewhere Principle* (EP)
If more than one L-tree can lexicalise the same S-tree (by the Superset Principle), then the L-tree with the least amount of superfluous material is chosen ('closest match')

- by the EP, all and only the roots of the *bujar*-type will be candidates to spell out QP

Limited Free Choice

(49) *Limited Free Choice* (LFC)
In the event of a tie, freely choose a candidate from the equally ranked ones, and insert it

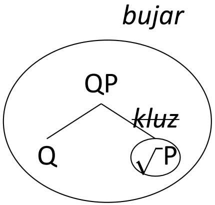
- by LFP, any one can be freely chosen from among the QP-sized ones
- larger roots (e.g. *star* 'old') still lose against QP-sized ones in the positive degree, and cannot get a spellout

Faithfulness

- suppose the syntax merges $\sqrt{}$P
  - SP/EP determines that all and only *kluz*-type roots can spell out $\sqrt{}$P
  - Limited Free Choice applies, inserting e.g. *kluz*
- the syntax goes on to merge QP
  - SP/EP determines that all and only *bujar*-type roots can spell out QP
  - Cyclic Override: the earlier spellout *kluz* will be overridden by an 'unfaithful' spellout

Faithfulness

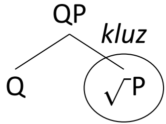
(50)



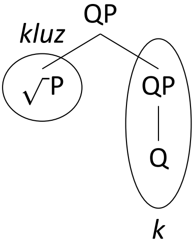
Faithfulness

- in order to derive the ‘faithful’ *kluz-k*, we need to apply a rescue strategy, and move the complement of Q, as in (52)
- by the SP, *-k* can spell out QP

(51)



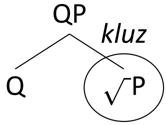
(52)



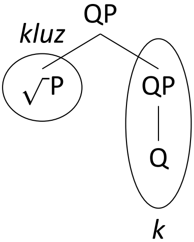
Faithfulness

- in order to derive the ‘faithful’ *kluz-k*, we need to apply a rescue strategy, and move the complement of Q, as in (52)
- by the SP, *-k* can spell out QP

(51)



(52)



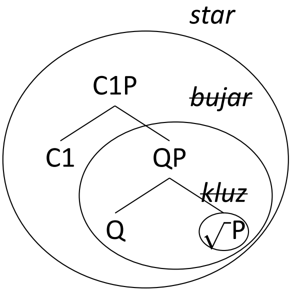
- however, the Spellout Algorithm blocks this derivation, since it favours nonmovement over movement derivations!

Faithfulness

- the same problem arises when the syntax goes on to merge C1
- two different root types are candidates for insertion by SP
  - *star*-type (C1P)
  - *ostr*-type (C2P)
- SP/EP determines that all and only *star*-type roots can spell out C1P
- Limited Free Choice applies
- Cyclic Override: the earlier spellout at QP (*bujar*) will be overridden

Faithfulness

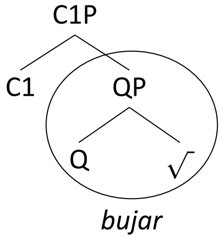
(53)



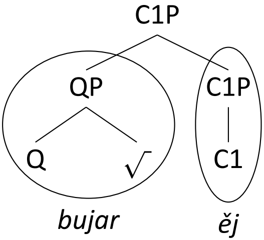
Faithfulness

- in order to derive *bujař-ej*, we need to apply a rescue strategy, and move the complement of C1
- by the SP, *-ěj* can spell out QP

(54)



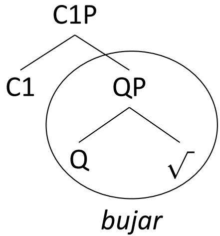
(55)



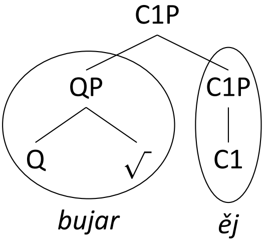
Faithfulness

- in order to derive *bujař-ej*, we need to apply a rescue strategy, and move the complement of C1
- by the SP, *-ěj* can spell out QP

(54)



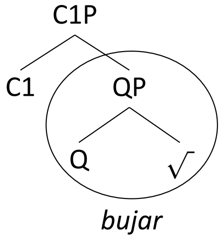
(55)



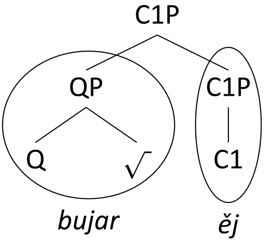
Faithfulness

- in order to derive *bujař-ej*, we need to apply a rescue strategy, and move the complement of C1
- by the SP, *-ěj* can spell out QP

(54)



(55)



- however, the Spellout Algorithm blocks this derivation, since it favours nonmovement over movement derivations!
- the productive allomorph *-ějš* is blocked by the existence of roots of C1P size!

Limited Free Choice

(56) *Unlimited Free Choice* (UFC)
At the first consultation of the lexicon, any type of root meeting the *Superset Principle* may be inserted

- the *Elsewhere Principle* is dispensed with
- this solves one of the competition problems, namely the fact that a bigger root (like *star* 'old') will lose against a smaller one (like *bujar* 'merry') in the positive degree

Faithfulness

- Cyclic Override is subject to a Faithfulness Restriction:

(57) *Faithfulness Restriction (FR)*
A spellout /α/ may override an earlier spellout /β/ iff
a. /α/ = /β/
b. /α/ contains a pointer to /β/

Faithfulness

- Cyclic Override is subject to a Faithfulness Restriction:

(57) *Faithfulness Restriction (FR)*
A spellout /α/ may override an earlier spellout /β/ iff
a. /α/ = /β/
b. /α/ contains a pointer to /β/

- the FR can be seen as a recoverability condition: cyclic override that is nonrecoverable is disallowed
  - if /α/ = /β/, override of /β/ by /α/ is recoverable
  - if /α/ contains a pointer to /β/, the content of /β/ is recoverable from /α/

Faithfulness

(57) *Faithfulness Restriction (FR)*

A spellout /α/ may override an earlier spellout /β/ iff

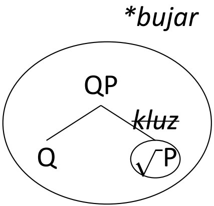
a. /α/ = /β/

b. /α/ contains a pointer to /β/

Faithfulness

- suppose *kluz* (size = $\sqrt{}$P) is inserted at $\sqrt{}$P
- the syntax goes on to merge QP
- nonmovement spellout at QP (e.g. *bujar*) is now blocked by FR, since /bujar/ would override /kluz/ (=unfaithful)

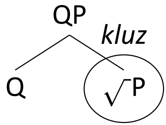
(58)



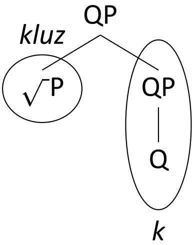
Faithfulness

- movement spellout has to be attempted to spell out Q

(59)



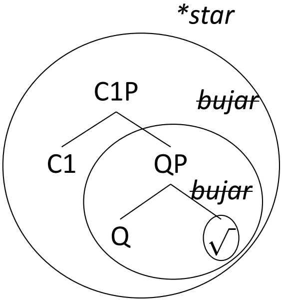
(60)



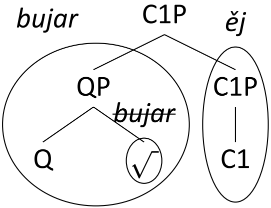
Faithfulness

- suppose *bujar* (size = QP) is inserted at $\sqrt{}$P (UFC, no EP!)
- the syntax goes on to merge QP
- nonmovement spellout at QP is allowed by FR, since /bujar/ overrides /bujar/
- at C1P, nonmovement spellout will violate FR, triggering movement

(61)



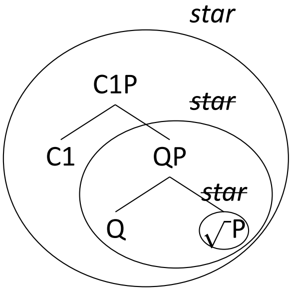
(62)



Faithfulness

- suppose *star* (size = C1P) is inserted at $\sqrt{}$P (UFC, no EP!)
- the syntax goes on to merge QP and C1P
- nonmovement spellout at QP and C1P are allowed by FR, since /star/ overrides /star/

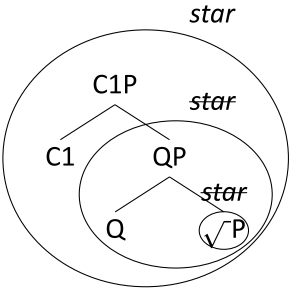
(63)



Faithfulness

- suppose *star* (size = C1P) is inserted at $\sqrt{}$P (UFC, no EP!)
- the syntax goes on to merge QP and C1P
- nonmovement spellout at QP and C1P are allowed by FR, since /star/ overrides /star/

(63)



- the same goes for roots of size C2P (e.g. *ostr*)

Faithfulness

(57) *Faithfulness Restriction (FR)*

A spellout /α/ may override an earlier spellout /β/ iff

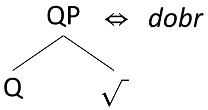
a. /α/ = /β/

b. /α/ contains a pointer to /β/

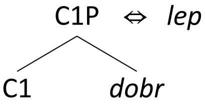
Faithfulness

- in cases of suppletion, FR allows cyclic override by a lexical item with a different phonology, provided it is lexically related to the lexical item being overridden
- lexical relatedness exists in virtue of pointers
- *lep*- ‘bett-’ contains a pointer to *dobr*- ‘good’ in its lexical entry

(64)

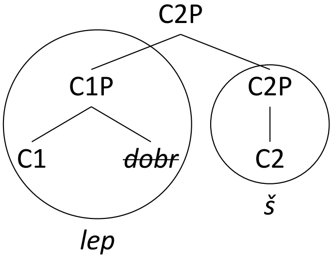


(65)



Faithfulness

(66)



- /lep/ can override /dobr/ since /lep/ contains a pointer to /dobr/
- /dobr/ is recoverable from /lep/ in virtue of its lexical relationship to /lep/

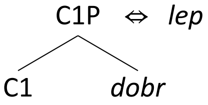
Faithfulness

A competition problem now arises

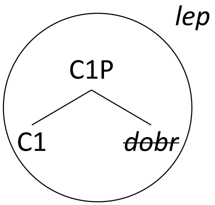
- at QP, we only want *dobr*- 'good' to be insertable, not *lep*- 'bett'
- SP as traditionally conceived selects both *dobr*- and *lep*- as candidates for insertion
- EP blocks *lep*- since *lep*- has more superfluous structure
- without EP, we need a different reason for blocking insertion of *lep*- at QP

Faithfulness

(67) *Restriction on Pointers* (RoP)
An item /α/ containing a pointer to /β/ can only be inserted if in the previous cycle /β/ was inserted

(68) 

- because of RoP, *lep*- 'bett-' cannot be inserted at QP
- only *dobr*- 'good' can get inserted at QP
- at C1P, *lep*- can be inserted since at the previous cycle *dobr* was inserted

(69) 

Summary

- the comparative consists of two heads (C1 and C2) instead of one CMPR head
- adjectives come in various sizes, spelling out different layers of functional structure
- in order to deal with the problem of Limited Free Choice, we abandon the Elsewhere Principle
- a Faithfulness Restriction on Cyclic Override is needed, restricting override as a function of recoverability

References

Starke, Michal. 2009. Nanosyntax: A short primer to a new approach to language. *Nordlyd* 36. 1–6.

Starke, Michal. 2018. Complex left branches, spellout, and prefixes. In Lena Baunaz, Karen De Clercq, Liliane Haegeman & Eric Lander (eds.), *Exploring nanosyntax*. 239–249. Oxford: Oxford University Press.