371.3::811.163.41'243
811.163.41'322
https://doi.org/10.18485/sj.2021.26.1.32

ДАНИЛО С. АЛЕКСИЋ*
БРАНИСЛАВА Б. ШАНДРИХ
Универзитет у Београду
Филолошки факултет

Оригинални научни рад
Примљен: 10. 11. 2020.
Прихваћен: 12. 1. 2021.

АУТОМАТСКА ЕКСЦЕРПЦИЈА ПАРОВА РЕЧИ ЗА УЧЕЊЕ ИЗГОВОРА У НАСТАВИ СРПСКОГ КАО СТРАНОГ ЈЕЗИКА

У раду се говори о минималним паровима из аспекта наставе српског као страног језика и из аспекта рачунарске лингвистике. Презентују се примена сегменталних минималних и сродних парова на једном курсу српског као страног језика и програм за аутоматску ексцерпцију таквих парова који је развио први аутор. Други аутор евалуира програм, указујући на његове добре особине и недостатке. Препоручује се употреба сегменталних минималних (и сродних) парова при вежбању изговора у настави српског као страног језика и износе се користи које наука може имати од написаног програма.

Кључне речи: минимални парови, српски као страни језик, глотодидактика, рачунарска лингвистика, корпусна лингвистика, фонетика, фонологија, *Python*.

0. У одабраним лингвистичким речницима, (фонолошки) минимални пар дефинише се као „dvije riječi ili dva morfema koji se razlikuju samo po jednom fonemu u jednakoj poziciji, a kod toga se ta razlika može svesti na nepodudaranje samo jedne diferencijalne oznake; npr. том – дом, кот – тот; e[ngleski] cat – cad, sin – sing, sink – zinc" (Симеон 1969: *s. v. par*; в. и „minimalni kontrastni par", *s. v. minimalan*), као „истраживачки поступак" и „две речи које се разликују по

* {danilo.aleksic, branislava.sandrih}@fil.bg.ac.rs

значењу када се промени само један глас", нпр. „*pin v. bin, cot v. cut*" (Кристал 2008: *s. v. minimal pair*) и, посредно, као лексички пар чији се чланови разликују само по испитиваном гласу: „*бак – бук, стол – стул, дом – том, папа – лапа, стал – стар*" (Тихонов и др. 2014: *s. v. мето́дика минима́льных пар*). Осим минималних парова, издвајају се и „минималне групе", попут „*big, pig, rig*" (Кристал 2008: *s. v. minimal pair*).

Чланови наведених минималних парова и наведене минималне групе диференцирају се на сегменталном (инхерентном) нивоу,[1] али постоје и супрасегментални минимални парови, тј. они чији се чланови међусобно супротстављају по акценту и/или постакценатској дужини. Пример би био „прозодијски минимални пар *барика́да:* ген. пл. *барика́да̄*" (Михаиловић 1973: 89, у коментару П. Ивића). У Барић и др. 1997: 53 експлицира се да сегментални минимални парњаци имају иста „naglasna svojstva".

М. Ивић (2001: 185) фонемску опозицију *б* ~ *п* илуструје речима *боб* и *поп*. Ако се минимални пар схвата као „две речи које се разликују по једној фонеми у једнакој позицији" – овде, једнаким позицијама – пар *боб* ~ *поп* јесте такав пар, јер су његови чланови сводиви на по две фонеме. Ако се пак сме заменити само један „глас", пар *боб* ~ *поп* не спада у минималне парове, јер се у речи *поп* уместо два гласа *б* јављају два гласа *п*.

Минимални парови се, према већ цитираним лексикографским изворима, у фонологији користе како би се утврдила опозиција између фонема (Симеон 1969: *s. v. komutacija*), открило који гласови припадају истој класи, или фонеми (Кристал 2008: *s. v. commutation* и *minimal pair*), односно установило које су гласовне опозиције („звуковые противопоставления") фонолошки релевантне, а које ирелевантне (Тихонов и др. 2014: *s. v. мето́дика минима́льных пар*). Другу примену у науци о функцијама гласова налази број минималних парова, који говори о функционалном оптерећењу дате опозиције („contrast") у проучаваном језичком систему (Кристал 2008: *s. v. function*). Зиндер (1970), међутим, сматра да се минимални парови у фонологији котирају превисоко.

1. Минимални парови се помињу и као материјал за учење изговора гласова из страног језика (Каваи,/Хиросе 2000; Демирезен 2005: 186–187, 190–191; Лу 2010; Маделска 2012: 32–38; Иљнер/Корнејева 2014: 113; Маирано/Калабро 2016). Вежбе са минималним паровима у настави енглеског као страног језика употребљавају се, чини се, бар од четрдесетих година XX века (Селсе Мурсија и др. 2006: 3–4). Сегментални минимални парови могу се срести у уџбеницима српског или српског и хрватског као страног језика објављеним у иностранству (Магнер 1998: 22, 37, 43, 49, 204; Марковић и др. 2002: 29, 33; Трофимкина/Дракулић Пријма 2012: 11; уп. Попова 1986: 26, 30, 183 и Коваљев

[1] У т. 0 пар *cad* ~ *cat* чита се као прозодијски једнообразан, а пар *cot* ~ *cut* на британском енглеском.

2000: 13), најчешће помешани са паровима који нису минимални, нпр. *брић ~ рић*, *већ ~ жеђ*, *са̏д ~ сат*; уџбеници Селимовић Момчиловић/Живанић 2008 и Милићевић Добромиров/Новковић 2009 и прилог *Изговор и писање гласова Ћ, Ч, Џ, Ђ* у приручнику за предаваче српског као страног језика *Српски с лакоћом* (Ломпар 2017) уз минималне сегменталне парове, као *ђак ~ џак*, *куће ~ куче* и *брати ~ прати*, садрже акценатски различите сегменталне парове, нпр. *сто ~ што*, *зевао ~ певао* (Милићевић Добромиров/Новковић 2009: 110) и *рећи ~ речи*. Сегментални минимални парови (ни они типа *сто ~ што*) нису примећени у Ћорић 1998[2], Бабић 2001[3], Ћорић/Никитовић 2005, Бјелаковић/Војновић 2006, Даниловић 2011[4], Бањац 2018 и Киш 2018.

Маирано и Калабро (2016: 264) наставницима саветују да вежбе са минималним паровима прилагоде вокабулару полазника̂, како непознатост значења не би склањала фокус са изговора и како би било јасно да је селектована фонолошка опозиција семантички значајна.

2. Двојица лингвиста аутори су софтвера за аутоматско прикупљање свих минималних парова са одређеном фонолошком опозицијом. У *Minimal Pair Finder* Бруса Хејза нужно је учитати листу речи или речник из којег ће се изоловати парови, по потреби проширени у групе („pat, tat, cat...”). Програм је на програмском језику *Visual Basic* 5, преузима се и инсталира (в. Л$_1$) и има једноставан графички интерфејс. Веб-апликација Паола Маирана *Minimal Pair Finder* (в. Л$_2$), са претраживачем на PHP-у (Маирано/Калабро 2016: 258), не допушта учитавање екстерне грађе. У оба програма укључено је успостављање парова по секвенцама, које нуди резултате попут енгл. *son ~ tin* и итал. *movimento ~ pavimento*.

Уместо по фонолошкој опозицији, парови се из *Clearpond*-а и *Worden*-а излиставају по унесеној речи или „не-речи” (Маирано/Калабро 2016: 258), нпр. *did*, *led*, *lit* итд. према инпуту *lid* у *Clearpond*-у.

3. Д. Алексић је користио сегменталне минималне и сродне парове у настави српског као страног језика и написао програм за лакше проналажење дотичних парова. Чланове парова не мора дистингвирати само једно фонолошко обележје и само једна реализација исте фонеме. Дакле, „минимални” су парови *ла̏к ~ лу̏к*, где се уочава разлика по грависности, дифузности и компактности (Симић/Остојић 1996: 178), и *ма̀гла ~ мо̀гло*, где је разлика у двама утеловљењима једне фонеме. Под „сродним” са сегменталним мини-

[2] Речи „*p*ût, *p*êt” на стр. 10 егземплификују консонант *n*, а не вокалску опозицију. Присутни су и примери „j*û*l, j*ûn*”, који су под „û”, итд. (в. стр. 8–11). Уп. Ћорић/Никитовић 2005: 6–11.

[3] Примери у контакту *лука* и *рука* (на стр. XIII и XVI) под *у* су. В. аналогне парове на стр. XV–XVI.

[4] Суседне речи „Džon” и „Džin” на стр. 3 при слову су *Dž*, а не код вокала̂.

малним подразумевају се парови *гѐл* ~ *гѐн*, *лантáна* ~ *лàнчанā* и сл., који би били минимални када би се игнорисала прозодија.

3.1. У другој недељи курса[5] одлучено је да се зађе у артикулацију гласова *ђ*, *љ*, *њ*, *ћ*, *ч* и *џ*, а као један од четири метода[6] одабрани су сегментални минимални и сродни парови. Већина примера за вежбу самостално је конструисана (неки су преузети из прилога В. Ломпар /2017: 40/).

На часу су најпре представљене анализе „C = TS", „Č = TŠ", „Ć = TSJ" и „Đ = DZJ", са надредним знаком „‿" изнад спојева десно од знака једнакости (које потичу из Белић 2006: 52–53). Наведене анализе праћене су указивањем на сличност графије и изговора код двојних слова *dž*, *lj* и *nj*. Затим је студентима дато упутство да при артикулацији што више збију („smash together") чланове комбинације *тс*, чланове комбинације *тш* итд. На супрасегменталном плану, вежбање је вођено тако што су неутралисане акценатске неподударности код „неминималних" парова. Иако су посреди били почетници, идеално би било да су сви парови били минимални, како би студенти од почетка курса били изложени стандардном српском прозодијском систему.

На часу је значење речи побудило интересовање нарочито када је било међујезичких поклапања (уп. Маирано/Калабро 2016: 264). Студенткињи из Кеније облик *чȕпā* био је занимљив због именице из свахилија *chupa* („флаша").

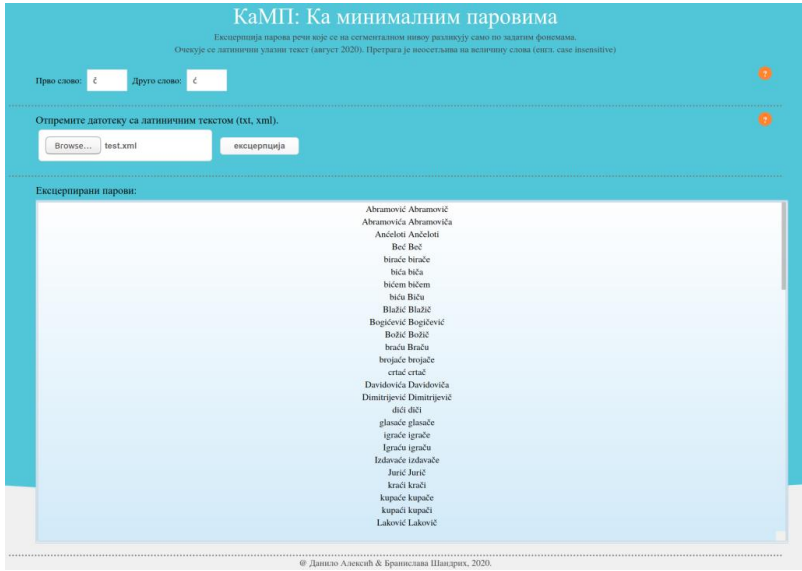
Позамашни фонд примера код полазника није изазвао досаду; на питање да ли као група желе да се окушају на свим паровима значајан број њих одговорио је потврдно.

3.2. Касније је пројектован и кодиран програм за аутоматско прикупљање сегменталних минималних и сродних парова. Провизорно именован *Ка минималним паровима* (скраћено КаМП; в. Прилог), програм је написан у програмском језику *Python* 3.8.2 (в. $Л_3$) и снабдевен онлајн сучељем (в. $Л_4$).[7]

[5] Саопштава се део искуства стеченог од фебруара до јула 2020. године у раду са групом од двадесет двоје студената из Азербејџана, Албаније, Гане, Еквадора, Есватинија, Јужног Судана, Кеније, Либана, Нигерије, Сијера Леонеа, Сирије, Суринама и Туркменистана и са Јамајке и Шри Ланке, на интензивном курсу српског као страног језика у оквиру пројеката *Свет у Србији* и *Србија за Србе из региона*. Само је полазница из Албаније имала предзнање српског језика. У време када је посебна пажња посвећивана изговору, овој групи је пет наставника држало по шест часова (од 45 минута) недељно.

[6] Уз разлагање африката и сонаната *љ* и *њ* на ине гласове и, у мањој мери, поређење са сличним гласовима у другим језицима и пружање података о положају који при артикулацији меких консонаната заузима врх језика (уп. нпр. Бабић 2001: XIV–XV и Марковић и др. 2002: 28).

[7] Онлајн сучеље је направила Б. Шандрих.



Слика 1. Интерфејс КаМП-а на $Л_4$

КаМП сегменталне минималне и сродне парове детектује, укратко, тако што из латиничког[8] српског улазног корпуса[9] црпе све речи које садрже граф (в. Кристал 2008: *s. v. graph*) прве испитиване фонеме, у свакој од тих речи граф прве испитиване фонеме супституише графом друге и проверава да ли је добијена реч забележена у истом корпусу. Конкретно, ако КаМП тражи опозицију *ћ* ~ *ч* и наилази на реч *seći*[10], он у њој замењује *ć* са *č* и проверава да ли нове речи има у истом корпусу. Уколико то јесте случај, КаМП на крају извршења наводи пар *seći* ~ *seči*. Ако КаМП при идентичном подешавању наилази и на реч *zaokrećući*, он у њој такође замењује *ć* са *č* и добијене облике (*zaokrečući*, *zaokrečući* и *zaokrećući*) тражи у улазном корпусу. Када они тамо нису посведочени, КаМП на крају извршења не наводи парове *zaokrećući* ~ *zaokrečući*, *zaokrećući* ~ *zaokrečući* и *zaokrećući* ~ *zaokrećući*.[11]

[8] Уз мање модификације, КаМП парове извлачи и из ћириличких корпуса, па и корпуса на другим писмима.

[9] Програм је са повољним исходом испробан на датотекама у форматима *txt* и *xml*. КаМП није за формате *doc*, *docx* и *pdf*.

[10] На латиници су примери који су тешње повезани са модусом рада КаМП-а и Хејзовог *Minimal Pair Finder*-а, јер Алексићев програм и, изгледа, *Minimal Pair Finder* захтевају латинички корпус.

[11] Смер процесирања (*ć* → *č* или *č* → *ć*) диктирале би фреквенције слова *č* и *ć* у корпусу.

Корисник треба да одабере сепаратор, елемент корпуса по којем се корпус може прелиминарно декомпоновати за учитавање тако да се не цепају речи. Ако је корпус подељен на параграфе, добар сепаратор је ознака за нови ред („\n”). Код корпусâ из једног пасуса, без иједне ознаке „\n”, сепаратор треба да буде неки други елемент који је релативно фреквентан у датом корпусу, попут белине између речи, тачке, запете итд., осим примакнуте цртице.

У резултујућим сегментима рêчи се проналазе помоћу регуларног израза (в. Фридл 2006) који враћа ниске домаћих и енглеских слова са селектованим словом или групом слова неокружене домаћим и енглеским словима, какве су подвучене ниске у реченици *Takve igrače* [не нпр. *igra*, због *č* – Д. А.] *publika voli* ако је у жижи слово *i*, односно подвучене ниске у наслову *„Gitar art” festival od 10. do 15. marta* ако је у жижи група слова *ar*. Јединице са примакнутом цртицом (*video-snimak* и сл.) третирају се као једна реч. Из аутпута се не би избацили скраћенички парови, нпр. *kg* ~ *km*, парови са спорним значењским диспаритетом, као *Mratindan* ~ *Mratinjdan* и *snegovi* ~ *snjegovi*, ни парови са спорним значењским диспаритетом и спорном нормативношћу једног или обају чланова, нпр. *Džeđang* ~ *Žeđang* (кин. 浙江, *Zhejiang*), где је само први члан (ћирилички) наведен у Пешикан и др. 2010, уз „трад.” *Чекјанг*.

Током обраде текста не разликују се велика и мала слова. Другим речима, КаМП би ексцерпирао парове *plivaču* ~ *Plivaću* и *Redži* ~ *ređi*.

КаМП игнорише дупликате, трипликате итд. парова у корпусу, без обзира на редослед чланова и величину слова. Након што је сузбијан сâм настанак дупликата у подацима који струје кроз програм, пар *zavesa* ~ *zavese* биће прескочен ако је већ регистрован пар *Zavese* ~ *zavesa*.

КаМП је бивао покренут на великим корпусима, и испрва је тада трошио много времена и меморије. Ово је утицало на избор начина на које се у КаМП-у постижу жељени циљеви, јер се настојало да се побољшају перформансе програма. Већу ефикасност омогућила је, између осталог, употреба генератора.

У коментарима у Прилогу (десно од хеша, „#”) назначени су текстови који су аутору помогли да отклони неке информатичке сумње и недоумице, не рачунајући званичну документацију. Најважнији су доприноси текста о стварању Декартових производа појединих слова у нискама (в. $Л_7$) и текста о генераторској функцији за парцелисано слање корпуса у радну меморију (в. $Л_{10}$).

У поређењу са Хејзовим *Minimal Pair Finder*-ом, КаМП је уопште експедитивнији и попустљивији је у погледу форме улазног корпуса, који не мора бити листа речи или одређена врста речника. КаМП подржава Unicode[12], док се у *Minimal Pair Finder* не могу унети слова попут *ж*, *џ*, *č* и *š* (уп. ипак Шипка 2005: 154–155). Потоњи програм увек разликује велико и мало слово,

[12] Тест са опозицијом *ћ* ~ *ч* на тексту у кодирању ISO-8859-1 није био успешан.

и зато нпр. пар *mera* ~ *Vera* види само када се унесу слова *m* и *V*, иако се она укуцавају у поља „Sound #1" и „Sound #2", а не „Letter #1" и „Letter #2". Било мана или не, изостављање сегменталних минималних и сродних група у алгоритму КаМП-а последица је ауторове одлуке.

Упити на Маирановом *Minimal Pair Finder*-у позивају минималне парове (не непогрешиво; в. $Л_2$), код већине понуђених језика паралелно са да́тима о фреквенцији чланова. Креирању и парсирању акцентованог српског корпуса свакако треба тежити, не само зарад екстракције минималних парова. Мада би КаМП био потпунији када би у аутпут додавао честоће обликâ у улазном корпусу, статистика не би била репрезентативна ако корпус не би био врло добро балансиран.

3.3. Прво што се да приметити током евалуације[13] КаМП-а, код је написан уредно и садржи темељно наведене спољашње изворе који су коришћени као узор или као документација. Програмска логика рашчлањена је на мање логичке целине у виду програмских функција, као што и добра пракса налаже. Битно је нагласити и да се предвиђа употреба великих улазних датотека, те при покретању, корпус се обрађује по сегментима, водећи притом рачуна о томе да сегментирање не утиче на неприродно прекидање реченица. Преоптерећење радне меморије се на тај начин избегава обрађивањем само оног сегмента корпуса који се разматра у датом тренутку. Функцијом *obrada_regeks_izraza()*, предвиђају се могуће словне грешке приликом писања диграфа. На пример, за слово *dž*, генерисан је регуларни израз који предвиђа све четири комбинације (DŽ/Dž/dŽ/dž). Начелно се детектују потенцијални кандидати (речи које садрже једно од два разматрана слова), генеришу се све комбинације замене посматраног слова са својим парњаком, а затим се тако добијене ниске претражују у истом корпусу.

Функцијом *zam_1_ili_više_sl()* генеришу се све могуће варијанте неке речи у којима се свако појављивање првог слова замени другим словом, али и обрнуто. На пример, за реч *чамчић* и слова *ћ* и *ч*, потенцијални парњаци су: *ћамћић*, *ћамћичи*, *ћамчић*, *ћамчичи*, *чамћић*, *чамћичи* и *чамчичи*. На овај начин, разматрају се сви могући кандидати, те се касније проверава њихова присутност у истом улазном тексту.

Потенцијалне модификације кода које би могле допринети његовој једноставности су ситне. Ради препознавања речи које садрже бар једно појављивање одређеног слова, користе се тзв. провере унапред и уназад (енгл. lookahead and lookbehind zero-length assertions). Нешто сложенији наведени регуларни израз могуће је поједноставити и заменити га регуларним изразом

[13] Евалуација КаМП-а запрема сегмент 3.3. Евалуатор је Б. Шандрих.

[A-Za-zĆ-ž-]*%s[A-Za-zĆ-ž-]*, тако да проналази исте речи. Иако нешто једноставнији за интерпретацију, нема значајни утицај на ефикасност извршавања.

Још једна мања измена могла би бити замена линија:

```
bafer_1 = prvo_slovo
bafer_2 = drugo_slovo
prvo_slovo = bafer_2
drugo_slovo = bafer_1
```

једноставним изразом за паралелну размену вредности променљивих (израз за тзв. *swap* алгоритам) који је специфичан за програмски језик Python:

```
prvo_slovo, drugo_slovo = drugo_slovo, prvo_slovo
```

Потенцијална ситна модификација јесте измештање назива улазне датотеке пре позивања помоћних функција, како би се избегла редундантност, али и смањила могућност грешке (на пример, приликом промене улазне датотеке). Добра пракса је користити релативно адресирање спољних датотека (нпр., уместо линије korpus = open(r"*C:\...\korpus.txt*"), написати линију korpus = open(r"*..\ulaz\korpus.txt*"), што би се тумачило као „вратити се један директоријум изнад директоријума у ком се налази код, позиционирати се у директоријум *ulaz/* који се ту налази, те отворити припадајућу датотеку *korpus.txt*", што не зависи од организације датотека нити оперативног система корисника).

Приликом испитивања, развијен је базични код који има исту сврху, без примене софистицираних метода оптимизације. Том приликом, на случајном узорку, не проналазе се парови које КаМП на истом узорку није и сам детектовао. Поставља се само питање важности величине слова. Примера ради, ако узорак за тестирање садржи парове (наведеним редоследом): *Igraću ~ igraču, Seća ~ seča, igraću ~ igraču, seća ~ seča*, КаМП проналази прва два пара, пошто при поређењу користи верзије речи написане малим словима (помоћу метода *casefold*). Ако се занемари писање великог слова, друга два пара јесу идентична са прва два. Због недовољне компетенције евалуатора у области лингвистике, ова појава неће бити даље дискутована.

4. Пошто је оруђе којим се сегментални минимални и сродни парови брзо и систематски просејавају из корпуса, КаМП је погодан за припрему говорних вежби,[14] уз тријажу парова по нивоу[15] (обичнији парови за ниже разине, одн. новински чланци и др. за Б1, белетристика, научни текстови и сл. за више нивое), по референцијској неједнакости и по нормативности (због „минималних парова" *волети ~ вољети*, *конфуционизам ~ конфучионизам*,

[14] Не само за странце (в. Бабић 1965).

[15] Идеју за ову инструкцију Алексић дугује проф. Весни Ломпар.

жуто-наранђастим ~ жуто-наранџастим, пара *ђеза* [„ђе за”] ~ *чеза* итд.[16]). Потенцијална грађа није мала, што демонстрира Табела 1.

Табела 1. Број парова по неким опозицијама у корпусу ПОЛ, без пречишћавања

Опозиција	Број сегменталних минималних и сродних парова у корпусу ПОЛ
ȣ ~ ђ	1.827
ђ ~ ћ	653
т ~ ћ	3.952
т ~ џ	4.378
т ~ ч	3.810
ћ ~ џ	2.721
ћ ~ ч	2.625

Корпус ПОЛ броји око 117.900.900 речи из 223.308 текстова са сајта *Политика*.

Поред методичке примене ту је и фонолошка – мерење функционалног оптерећења (Маирано/Калабро 2016: 255). Према О и др. 2015: 153, када се „свим фонемама у инвентару прида исти значај, уз занемаривање њихове фреквенције и њихове улоге у опозицијама [contrasts], одређени кључни феномени остају прениско вредновани”.

Дериватолог располаже вађењем парова по опозицијама *ира ~ иса*/*ише*, *супер ~ хипер* (уп. Бабић 2002: 500) итд.

Са лексикографског становишта, КаМП би олакшао израду српског речника минималних парова. Усавршени КаМП, осетљив на прозодијске особине, са специјалним корпусом, учинио би је излишном.

5. На основу литературе и праксе, препоручује се примена минималних (и сродних) парова са иним поступцима у настави изговора српског као страног језика, а то навешћује прву конклузију чланка. Друга конклузија гласи да је предавачима српског као страног језика бављење рачунарском лингвистиком плодоносна активност. Најзад, може се поновити оцена да је КаМП солидан алат за ексцерпцију сегменталних минималних и сродних парова.

[16] Уп. Попова 1986: 174.

ЛИТЕРАТУРА

Бабић 1965: S. Babić, Osnova za metodsku obradu glasova č i ć, Zagreb: *Jezik*, 13/3, 83–95.

Бабић [5]2001: S. Babić, *Serbian/Croatian for foreigners*. *Book 1*, Beograd: Zadužbina Ilije M. Kolarca.

Бабић [3]2002: S. Babić, *Tvorba riječi u hrvatskome književnome jeziku*, Zagreb: Globus.

Бањац [6]2018: P. Banjac, *Serbian for foreigners*, Beograd: Raška škola.

Барић и др. [2]1997: E. Barić, M. Lončarić, D. Malić, S. Pavešić, M. Peti, V. Zečević, M. Znika, *Hrvatska gramatika*, Zagreb: Školska knjiga.

Белић [2]2006: А. Белић, *Историја српског језика*, Београд: Завод за уџбенике и наставна средства.

Бјелаковић/Војновић [2]2006: I. Bjelaković, J. Vojnović, *Научимо српски*. *1*, Novi Sad: Filozofski fakultet, Dnevnik.

Даниловић 2011: M. Danilović, *Step by Step Serbian. 1*, Beograd: Kornet.

Демирезен 2005: Demirezen, M. Rehabilitating a Fossilized Pronunciation Error: the /v/ and /w/ Contrast by Using the Audio Articulation Method in Teacher Training in Turkey. *Journal of Language and Linguistic Studies*, 1/2. ‹https://www.jlls.org/index.php/jlls/article/view/15›. 15.06.2020. 183–192.

Зиндер 1970: Л. Р. Зиндер, О «минимальных парах», *Язык и человек*, Москва: Издательство Московского университета, 105–109.

Ивић [9]2001: M. Ivić, *Pravci u lingvistici*. *I*, Beograd: Čigoja štampa.

Ильнер/Корнејева 2014: А. О. Ильнер, Л. И. Корнеева, Основные положения методики развития речевого слуха у студентов неязыковых специальностей на начальном этапе обучения иностранным языкам, *Известия Уральского федерального университета. Сер. 1*. *Проблемы образования, науки и культуры*, 123/1, Екатеринбург, 108–115.

Каваи/Хиросе 2000: G. Kawai, K. Hirose, Teaching the pronunciation of Japanese double-mora phonemes using speech recognition technology, *Speech Communication*, 30/2–3, 131–143.

Киш [3]2018: J. Kiš, *Step into Serbian*. *Serbian for foreigners*, Beograd: Službeni glasnik.

Ковальев 2000: Н. С. Ковалев, *Современный сербский язык*, Волгоград: Издательство Волгоградского государственного университета.

Кристал ⁶2008: D. Crystal, *A Dictionary of Linguistics and Phonetics*, Malden, MA: Blackwell Publishing.

Л$_1$: B. Hayes. *Minimal Pair Finder*. ‹https://linguistics.ucla.edu/people/hayes/103/MinimalPairs/index.htm›. 13.06.2020.

Л$_2$: P. Mairano. *Minimal Pair Finder*. ‹http://phonetictools.altervista.org/minimalpairfinder›. 13.06.2020.

Л$_3$: Python Software Foundation. *Python*. ‹https://www.python.org›. 13.06.2020.

Л$_4$: Д. Алексић, Б. Шандрих. *КаМП: Ка минималним паровима*. ‹http://kamp.jerteh.rs›. 10.09.2020.

Л$_5$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/60634040›. 13.06.2020.

Л$_6$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/11144539›. 22.06.2020.

Л$_7$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/52383460›. 13.06.2020.

Л$_8$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/1155647›. 13.06.2020.

Л$_9$: Чланак на сајту. ‹https://www.w3schools.com/python/ref_string_count.asp›. 13.06.2020.

Л$_{10}$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/30775393›. 10.09.2020.

Л$_{11}$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/51976543›. 30.09.2020.

Л$_{12}$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/152596›. 18.09.2020.

Л$_{13}$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/209854›. 13.06.2020.

Л$_{14}$: Одговор на сајту *Stack Overflow*. ‹https://stackoverflow.com/a/28666854›. 18.09.2020.

Ломпар 2017: В. Ломпар, Изговор и писање гласова Ћ, Ч, Џ, Ђ, у: В. Ломпар (прир.), *Српски с лакоћом. Приручник за предаваче српског језика као страног*, Београд: Међународни славистички центар, 39–42.

Лу 2010: T. T. Luu, Teaching English Discrete Sounds through Minimal Pairs, *Journal of Language Teaching and Research*, 1/5, 540–561.

Магнер 1998: T. Magner, *Introduction to the Croatian and Serbian Language. Revised Edition*, University Park, Pennsylvania: The Pennsylvania State University Press.

Маделска 2012: L. Madelska, *Практическая грамматика польского языка*, Kraków: Universitas.

Маирано/Калабро 2016: P. Mairano, L. Calabrò, Are minimal pairs too few to be used in pronunciation classes?, in: R. Savy, I. Alfano (eds.), *La fonetica sperimentale nell'insegnamento e nell'apprendimento delle lingue straniere. Phonetics and language learning*, Milano: Officinaventuno, 255–268.

Марковић и др. ²2002: М. Маркович, Е. Йоканович-Михайлова, М. П. Киршова, В. Н. Зенчук, *Сербский язык*, Москва: МУБиУ.

Милићевић Добромиров/Новковић ³2009: Н. Милићевић-Добромиров, Б. Новковић, *Учимо српски 1*, Нови Сад: Азбукум.

Михаиловић 1973: Љ. Михаиловић, Фонолошки принцип у српскохрватском правопису после Вука Караџића, Београд: *Научни састанак слависта у Вукове дане*, 3, 85–91.

О и др. 2015: Y. M. Oh, C. Coupé, E. Marsico, F. Pellegrino, Bridging phonological system and lexicon: Insights from a corpus study of functional load, *Journal of Phonetics*, 53, 153–176.

Пешикан и др. 2010: М. Пешикан, Ј. Јерковић, М. Пижурица, *Правопис српскога језика*, Нови Сад: Матица српска.

Попова 1986: Т. П. Попова, *Сербскохорватский язык*, Москва: «Высшая школа».

Селимовић Момчиловић/Живанић ⁷2008: М. Селимовић-Момчиловић, Љ. Живанић, *Српски за странце. Реч по реч*, Београд: Институт за стране језике.

Селсе Мурсија и др. 2006 [1996]: M. Celce-Murcia, D. Brinton, J. Goodwin, *Teaching Pronunciation. A Reference for Teachers of English to Speakers of Other Languages*, New York: Cambridge University Press.

Симеон 1969: R. Simeon, *Enciklopedijski rječnik lingvističkih naziva*, Zagreb: Matica hrvatska.

Симић/Остојић ³1996: Р. Симић, Б. Остојић, *Основи фонологије српског књижевног језика*, Београд: Универзитет у Београду.

Тихонов и др. ²2014: А. Н. Тихонов, Р. И. Хашимов, Г. С. Журавлева, М. А. Лапыгин, А. М. Ломов, Л. В. Рацибурская, Е. Н. Тихонова, *Энциклопедический словарь-справочник лингвистических терминов и понятий*. *Русский язык*, Москва: ФЛИНТА.

Трофимкина/Дракулић Прийма ²2012: О. И. Трофимкина, Д. Дракулич-Прийма, *Сербский язык*. *Начальный курс*, Санкт-Петербург: КАРО.

Ћорић ⁶1998: Б. Ћорић, *Српски за странце*, Београд: Чигоја штампа.

Ћорић/Никитовић 2005: Б. Ћорић, З. Никитовић, *Српски језик за странце*, Источно Сарајево: Завод за уџбенике и наставна средства.

Фридл ³2006: J. Friedl, *Mastering Regular Expressions*, Sebastopol, CA: O'Reilly Media, Inc.

Шипка 2005: D. Šipka, Leksičko opterećenje opozicije zvučni/bezvučni u tri slovenska jezika, Нови Сад: *Зборник Матице српске за славистику*, 68, 153–160.

AUTOMATIC EXCERPTION OF WORD PAIRS FOR PRONUNCIATION LEARNING IN TEACHING OF SERBIAN AS A FOREIGN LANGUAGE

Summary

The paper discusses minimal pairs from the perspectives of teaching of Serbian as a foreign language and of computational linguistics. The application of segmental minimal and similar pairs in a course of Serbian as a foreign language and a program for automatic excerption of such pairs developed by the first author are presented. The second author evaluates the program, pointing out its good features and shortcomings. Use of segmental minimal (and similar) pairs during the pronunciation exercises in teaching of Serbian as a foreign language is recommended, and the scientific benefits of the program are listed.

Keywords: minimal pairs, Serbian as a foreign language, glottodidactics, computational linguistics, corpus linguistics, phonetics, phonology, *Python*.

Danilo S. Aleksić
Branislava B. Šandrih

Прилог. Програм *Ка минималним паровима* (КаМП)

```
from itertools import product
import re
import sys
sys.stdout.reconfigure(encoding = "utf-8")    # В. Л5.
prvo_slovo = "đ".casefold()
drugo_slovo = "dž".casefold()

def spajanje_niski(*niske):
   return "".join(niske)

def obrada_regeks_izraza(slovo):
   if len(slovo) == 1:
      regeks_izraz = spajanje_niski(
            "[", slovo.upper(), slovo, "]")
      return regeks_izraz
   else:
      lista_za_regeks_izraz = ["("]
      for oblik in map("".join, product(*((
            karakter.upper(), karakter.lower())
            for karakter in slovo))):    # В. Л6.
         lista_za_regeks_izraz.append(spajanje_niski(
               oblik, "|"))
      lista_za_regeks_izraz.append(")")
      regeks_izraz = "".join(lista_za_regeks_izraz).replace(
            "|)", ")")
      return regeks_izraz

def zamena_jednog_slova(reč):
   reč_malim_slovima_sa_zamenama = reč.replace(
         prvo_slovo, drugo_slovo)
   lista_reči_malim_slovima_sa_zamenama = []
   lista_reči_malim_slovima_sa_zamenama.append(
         reč_malim_slovima_sa_zamenama)
   return lista_reči_malim_slovima_sa_zamenama

def zam_1_ili_više_sl(reč):
   lista_reči_malim_slovima_sa_zamenama = []
   reč_za_obradu = reč
   reč_za_obradu = reč_za_obradu.replace(prvo_slovo, "{}")
   reč_za_obradu = reč_za_obradu.replace(drugo_slovo, "{}")
   for kombinacija in product(
```

```
            [prvo_slovo, drugo_slovo],
            repeat = reč_za_obradu.count("{}")):    # Уп. Л7. В. Л8 и Л9.
        reč_malim_slovima_sa_zamenama = reč_za_obradu.format(
                *kombinacija)
        if reč_malim_slovima_sa_zamenama != reč:
            lista_reči_malim_slovima_sa_zamenama.append(
                    reč_malim_slovima_sa_zamenama)
    return lista_reči_malim_slovima_sa_zamenama

def segmentacija_korpusa(korpus, separator = "\n"):    # Уп. Л10.
    red = ""
    while True:
        komad = korpus.read(8192)
        if komad == "":
            yield red
            break
        while True:
            i = komad.rfind(separator)
            if i == -1:
                break
            yield spajanje_niski(red, komad[:i])
            red = ""
            komad = komad[i+1:]
        red = spajanje_niski(red, komad)

def obrada_prvog_slova(prvi_diferencijalni_izraz):
    lista_reči_malim_slovima_sa_zamenama = []
    skup_taplova_sa_zamenama = set()
    korpus = open(r"C:\...\korpus.txt", "r", encoding = "utf-8")
    komadi = segmentacija_korpusa(korpus)
    for komad in komadi:
        komad = komad.replace("\u00ad", "")    # В. Л11.
        pogoci_sa_prvim_slovom = re.finditer(
                f"(?<![A-Za-zĆ-ž])([A-Za-zĆ-ž-]*{prvi_diferencijalni_izraz}"
                "[A-Za-zĆ-ž-]*)(?![A-Za-zĆ-ž])", komad)
        for pogodak in pogoci_sa_prvim_slovom:
            reč = pogodak.group(0)
            reč = reč.strip("-")
            if isinstance(reč, tuple):    # В. нпр. Л12.
                reč = reč[0]
            reč_malim_slovima = reč.casefold()
```

```
        if prvo_slovo in drugo_slovo or drugo_slovo in prvo_slovo:
            broj_preklapanja = (prvo_slovo.count(drugo_slovo)
                    + drugo_slovo.count(prvo_slovo))
            if drugo_slovo in reč_malim_slovima:
                if (reč_malim_slovima.count(prvo_slovo)
                        + reč_malim_slovima.count(drugo_slovo)
                        == broj_preklapanja + 1):
                    lista_reči_malim_slovima_sa_zamenama = zamena_jednog_slova(
                            reč_malim_slovima)
                else:
                    lista_reči_malim_slovima_sa_zamenama = zam_1_ili_više_sl(
                            reč_malim_slovima)
            else:
                if reč_malim_slovima.count(prvo_slovo) == 1:
                    lista_reči_malim_slovima_sa_zamenama = zamena_jednog_slova(
                            reč_malim_slovima)
                else:
                    lista_reči_malim_slovima_sa_zamenama = zam_1_ili_više_sl(
                            reč_malim_slovima)
        else:
            if (reč_malim_slovima.count(prvo_slovo)
                    + reč_malim_slovima.count(drugo_slovo) == 1):
                lista_reči_malim_slovima_sa_zamenama = zamena_jednog_slova(
                        reč_malim_slovima)
            else:
                lista_reči_malim_slovima_sa_zamenama = zam_1_ili_više_sl(
                        reč_malim_slovima)
        if lista_reči_malim_slovima_sa_zamenama:
            tapl_sa_zamenama = tuple(
                    lista_reči_malim_slovima_sa_zamenama)
            tapl_za_autput = reč, tapl_sa_zamenama
            if tapl_za_autput not in skup_taplova_sa_zamenama:
                skup_taplova_sa_zamenama.add(tapl_za_autput)
                yield tapl_za_autput

def obrada_drugog_slova(drugi_diferencijalni_izraz):
    brojač = set()
    izl_zip_lista = []
    korpus = open(r"C:\...\korpus.txt", "r", encoding = "utf-8")
    komadi = segmentacija_korpusa(korpus)
    for komad in komadi:
        komad = komad.replace("\u00ad", "")
```

```
        reči_sa_drugim_slovom = re.findall(
            f"(?<![A-Za-zĆ-ž])([A-Za-zĆ-ž-]*{drugi_diferencijalni_izraz}"
            "[A-Za-zĆ-ž-]*)(?![A-Za-zĆ-ž])", komad)
        if len(drugo_slovo) == 1:
            reči_sa_drugim_slovom = [
                reč.strip("-") for reč in reči_sa_drugim_slovom]
            reči_sa_drugim_slovom_malim_slovima = [
                reč.casefold()
                for reč in reči_sa_drugim_slovom]
        else:
            reči_sa_drugim_slovom = [
                reč[0].strip("-")
                for reč in reči_sa_drugim_slovom]
            reči_sa_drugim_slovom_malim_slovima = [
                reč.casefold()
                for reč in reči_sa_drugim_slovom]
        zip_iterator = zip(
            reči_sa_drugim_slovom,
            reči_sa_drugim_slovom_malim_slovima)    # Уп. Л13.
        for prva_vrednost, druga_vrednost in zip_iterator:
            if druga_vrednost not in brojač:
                brojač.add(druga_vrednost)
                izl_zip_lista.append((prva_vrednost, druga_vrednost))
    return izl_zip_lista
prvi_skup_parova = set()
drugi_skup_parova = set()
x = 0
y = 0
korpus = open(r"C:\...\korpus.txt", "r", encoding = "utf-8")
komadi = segmentacija_korpusa(korpus)
for komad in komadi:
    komad_za_brojanje = komad.casefold()
    x = x + komad_za_brojanje.count(prvo_slovo)
    y = y + komad_za_brojanje.count(drugo_slovo)
if x > y:
    bafer_1 = prvo_slovo
    bafer_2 = drugo_slovo
    prvo_slovo = bafer_2
    drugo_slovo = bafer_1
izlazni_taplovi = obrada_prvog_slova(obrada_regeks_izraza(prvo_slovo))
izlazna_zip_lista = obrada_drugog_slova(obrada_regeks_izraza(drugo_slovo))
```

```
for tapl in izlazni_taplovi:
   if len(tapl[1]) == 1:
      for prva_vrednost, druga_vrednost in izlazna_zip_lista:
         if "".join(tapl[1]) == druga_vrednost:    # В. нпр. Л14.
            if (("".join(tapl[0]).casefold(), prva_vrednost.casefold())
                  not in [(par[0].casefold(), par[1].casefold())
                  for par in prvi_skup_parova]):
               prvi_skup_parova.add(("".join(tapl[0]), prva_vrednost))
   else:
      for član in tapl[1]:
         for prva_vrednost, druga_vrednost in izlazna_zip_lista:
            if član == druga_vrednost:
               if (("".join(tapl[0]).casefold(), prva_vrednost.casefold())
                     not in [(par[0].casefold(), par[1].casefold())
                     for par in prvi_skup_parova]):
                  prvi_skup_parova.add(("".join(tapl[0]), prva_vrednost))
for par in prvi_skup_parova:
   if (spajanje_niski(par[1], " ~ ", par[0])
         not in drugi_skup_parova):
      drugi_skup_parova.add(spajanje_niski(par[0], " ~ ", par[1]))
lista_parova = list(drugi_skup_parova)
lista_parova.sort(key = str.lower)
for član in lista_parova:
   print(član)
korpus.close()
print("\n\tBROJ PAROVA:")
print("\t\t", len(lista_parova))
```