

Милош Утвић

Информатички практикум 4

Регуларни изрази

Азбука. Ниске

- ▶ У општем случају азбука представља произвољан непразан скуп симбола, а ниска над том азбуком је коначан низ симбола те азбуке.
- ▶ Број карактера ниске зваћемо **дужином ниске**. Приметимо да се сваки карактер може посматрати и као карактер и као **ниска дужине 1**.
- ▶ Према овој дефиницији речи српског језика су ниске над његовом азбуком (ћириличном, латиничном или унијом обе). Исто важи и за остале природне језике. Међутим, ниске могу бити и хемијске и математичке формуле, као и разни други вештачки језици.
- ▶ На пример, ако је дата трочлана азбука $A = \{ @, \#, ! \}$, по датој дефиницији ниске над том азбуком којима се обично означавају псовке у стриповима су:
@, #, ! (ниске дужине 1),
@@, @#, @!, #@, ##, #!, !@, !#, !! (ниске дужине 2),
@@@, @@#, @@!, @#@, @##, @#!, @!@, @!#, @!! (ниске дужине 3),
@#!@ (пример ниске дужине 4)
итд.
- ▶ Чак се и бројеви могу посматрати као ниске над одређеном азбуком.

Бројеви као ниске

- ▶ Природни бројеви се могу посматрати као ниске над азбуком $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ које почињу цифром која није нула: 7, 16, 987, 2378 итд.
- ▶ Цели бројеви се могу посматрати као ниске над азбуком $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -\}$ које су:
 - ▶ или једнаке 0
 - ▶ или почињу цифром која није нула (7, 16, 987, 2378 итд.)
 - ▶ или почињу симболом знака (+ или -) и цифром која није нула (-236, -89, 0, 7, 16, 987, 2378 итд.).
- ▶ Децимални бројеви се могу посматрати као као ниске над азбуком целих бројева допуњеном запетом, при чему важе иста ограничења као код целих бројева уз додатак да се запета обавезно појављује само једном у запису ниске и увек раздваја две групе цифара (цео део и децимале): 0,34 -16,32 +289,65 итд.

Напомене. (Unicode-)ниске

- ▶ У претходним дефиницијама треба обратити пажње на термине **скуп** и **низ**. Скуп не подразумева никакву уређеност елемената, па је азбука (скуп симбола) у општем случају неуређена.
- ▶ С друге стране, термином **низ** у дефиницији ниске се истиче да ниска није обичан већ **уређен** скуп карактера у смислу да је битно на којој позицији се налази који карактер. Према томе, различит редослед карактера одређује различите ниске. На пример, ниске **лук** и **кул** нису исте, иако су у питању исти скупови карактера ($\{\text{л}, \text{у}, \text{к}\} = \{\text{к}, \text{у}, \text{л}\}$).
- ▶ Ако ограничимо азбуку на карактерски скуп Unicode, ниске над том азбуком називаћемо **Unicode-нискама**.
- ▶ У даљем тексту, ако се не каже другачије, увек се подразумева да је **азбука** комплетан карактерски скуп **Unicode**, а **ниска** искључиво **Unicode-ниска**, тј. коначан низ Unicode-карактера неког текста.

Интерпретација ниске

- ▶ Иако по својој дефиницији делује да је ниска заправо математичка апроксимација за текстуалну реч, она је много више од тога. Ако, на пример, за азбуку узмемо српску ћирилицу са уобичајеним знацима интерпункције и белинама (размаци и знаци за нови ред), видимо да ниска може бити било који структурни елемент текста:
 - ▶ слово, односно једнословна реч (*Б*),
 - ▶ произвољна текстуална реч (*Београд*)
 - ▶ вишечлана лексема (*Нови Сад*)
 - ▶ синтагма (*леји њлави Дунав*)
 - ▶ једна линија текста (на пример стих *Кад је њао мрак*)
 - ▶ реченица (*Нови Сад је град на лејом њлавом Дунаву.*)
 - ▶ а такође и пасус, одељак, глава, произвољан део текста или комплетан текст.
- ▶ У пракси, приликом аутоматске обраде текста, ниска може да има било коју од горњих интерпретација.

Регуларни изрази (енгл. Regular Expressions, скр. РЕ)

- ▶ Регуларни изрази (РЕ) су специјалне ниске које се користе као обрасци/формуле, односно скраћена ознака за некакав скуп ниски (коначан или бесконачан).
- ▶ При томе је немогуће направити регуларан израз за произвољан скуп ниски, већ само за једну класу скупова који се називају регуларни језици.
- ▶ На пример, било који коначан скуп је регуларан језик и може се описати регуларним изразом. Међутим, само за неке бесконачне скупове постоји регуларни израз који их описује.

Карактери као регуларни изрази

- ▶ Највећи број Unicode-карактера су најједноставнији могући регуларни изрази који представљају сами себе.
- ▶ На пример, мало ћирилично слово **а** је регуларан израз који представља мало ћирилично слово а. Слично важи и за остала ћирилична слова српске азбуке, слова енглеске абецеде и слова осталих азбука природних језика.
- ▶ Било која арапска цифра (од **0** до **9**) представља саму себе (**0** је цифра нула, **9** је цифра девет итд.).
- ▶ Кад су у питању знаци интерпункције, већина њих представљају сами себе, али постоји и значајан број изузетака који се називају **метакарактери регуларних израза**.

Дописивање ниски (и карактера)

- ▶ Главна операција над нискама једне азбуке јесте дописивање. Та операција обично нема ознаку јер је резултат очигледан.
- ▶ На пример, дописивањем двеју ниски, **час** и **ови**, добија се ниска **часови**.
- ▶ Јасно је да се свака ниска може представити као резултат вишеструких дописивања њених појединачних карактера у одређеном редоследу. Искористићемо заграде како бисмо нагласили редослед извођења операције дописивања:

часови = (((ча)с)о)в)и

Дописивање регуларних израза

- ▶ На исти начин се могу дописивати и регуларни изрази. У случају када регуларни изрази који се дописују представљају обичне карактере, резултат дописивања је и ниска и регуларан израз који представља ту ниску.
- ▶ Дакле, **часови** је и ниска и регуларан израз који представља ту ниску.
- ▶ Према томе, најједноставнији регуларни изрази (РЕ) су обични карактери и ниске настале њиховим дописивањем и они представљају сами себе, тј. **једну ниску** (у случају карактера та ниска је дужине 1).
- ▶ Оно што омогућава да РЕ означава више од једне ниске јесу метакарактери и операције са регуларним изразима.

Метакараактери регуларних израза

- ▶ **Метакараактери су** посебни карактери који се не користе у свом уобичајеном значењу, тј. не представљају сами себе, већ имају специјално значење.
- ▶ Скуп метакарактера није фиксиран већ зависи од конкретне имплементације регуларних израза. На пример, регуларни изразе имплементирају Notepad++, Python, JavaScript, PHP и XML Schema, али не мора да значи да ће користити исти скуп метакарактера, нити да ће исти метакараактери имати иста значења у различитим имплементацијама.

Метакараактери програма Notepad++

- ▶ Овде ћемо анализирати метакараактере које подржава Notepad++ и њихове улоге:
 - ▶ Груписање: обичне заграде као метакараактери, односно (и);
 - ▶ Алтернација или унија: метакараактер | (вертикална црта);
 - ▶ Џокер знак: метакараактер . (тачка);
 - ▶ Карактерске класе: метакараактери [] ^ (средње заграде и капица);
 - ▶ Метакараактери квантификације: * + ? { } (звездица, плус, упитник, витичасте заграде);
 - ▶ Контекстни метакараактери или сидра: ^ \$ (капица, долар) и
 - ▶ Врачара: \ (обрнута коса црта, енгл. *backslash*).

Примена регуларних израза у програму Notepad++

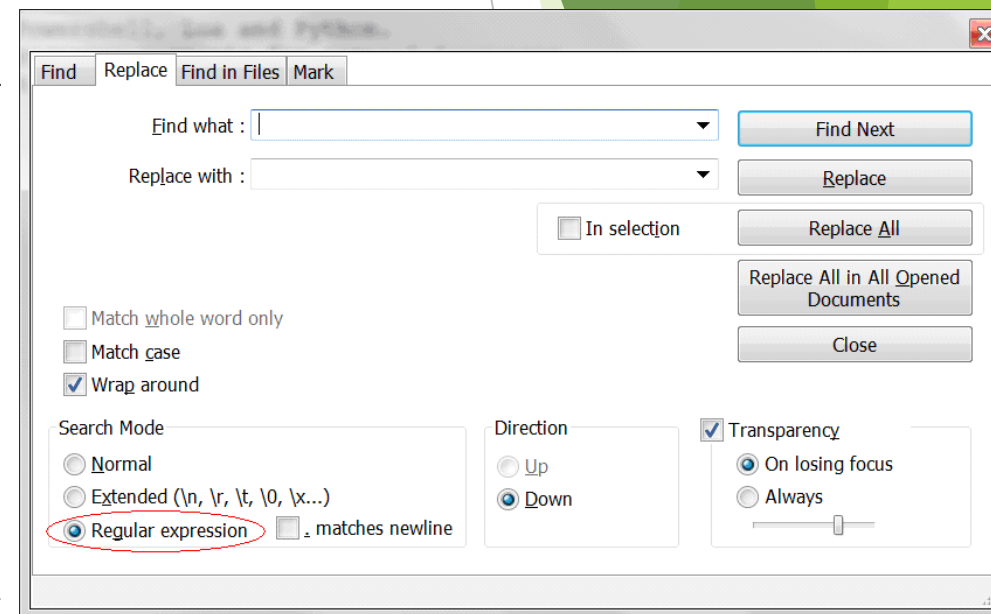
- Пре сваке претраге или замене у програму Notepad++ потребно је проверити да ли је у дијалогу *Find/Replace* изабрана опција *Regular expressions*. У противном их програм третира као обичне ниске.

- Надаље ће ознаке

Find: *шражениТекст*

Replace: *текстЗамене*

означаваати ниске које треба унети у поља дијалога *Find* (**регуларни израз који се тражи означен црвеном бојом**) и *Replace* (**текст замене означен плавом бојом**). Тиме избегавамо неспоразума да ли нека белина представља део регуларног израза или не,



Примери употребе регуларних израза (| = „или“, унија, алтернација)

- ▶ Пронађимо у тексту све ћириличне верзије појављивања речи **дошао** (дошао, дошо, дош'о).
- ▶ I начин: користимо унију (алтернацију) ниски
Find: **дошао|дошо|дош'о**
- ▶ II начин: користимо унију и могућност скраћеног записа:
Find: **дош(ао|о|'о)**
- ▶ Приметимо да се дописивање и унија „слажу“ као множење и сабирање. Наиме, као што важи $2 \cdot (3 + 4) = 2 \cdot 3 + 2 \cdot 4$, тако важи да је **дош(ао|о|'о)** исто што и **дошао|дошо|дош'о**. Скраћени запис **дош(ао|о|'о)** можемо да прочитамо и као: низ карактера **дош** на које се дописује или **ао** или **о** или **'о**.

Груписање и приоритет

- ▶ Поставља се питање: да ли су и зашто у регуларном изразу $\text{дош}(\text{ао}|\text{о}'\text{о})$ неопходне заграде? Ако пажљиво анализирамо, видимо да се у изразу „сударају“ две операције: дописивање и алтернација (унија). Када напишемо израз $\text{а}\delta|\text{ц}$, није јасно која операција се прва извршава, тј. израз може да се интерпретира на два начина:
 1. $\text{а}(\delta|\text{ц})$, тј. прво се обавља алтернација, па онда дописивање (као резултат се препознају ниске $\text{а}\delta$ и ац) или
 2. $(\text{а}\delta)|\text{ц}$, тј. прво се обавља дописивање, па онда унија (као резултат се препознају ниске $\text{а}\delta$ и ц).
- ▶ Овај проблем се решава увођењем **приоритета** операција: увек ће дописивање имати већи приоритет (тј. прво се извршити) у односу на алтернацију (унију).

Груписање и приоритет (2)

- Према томе, да бисмо избегли погрешну интерпретацију, морамо користити заграде за избегавање приоритета операција јер операција у заградама увек има највећи приоритет:

дош(ао|о|'о)

Увођењем заграда сада се прво извршава алтернација (унија), па тек онда дописивање.

- У противном би дошао|о|'о имао исто значење као и дошао)|о|'о, тј. препознавао би ниске дошао, о и 'о.

Комбиновање уније (алтернације) и дописивања (1)

- Пронађимо у тексту сва појављивања (ћирилична и латинична) речи **дошао** (дошао, дошо, дош'о, došao, došo, doš'o).

- I начин: користимо унију (алтернацију) ниски

Find: **дошао|дошо|дош'о|došao|došo|doš'o**

- II начин: користимо унију и могућност скраћеног записа:

Find: **дош(ао|о|'о)|doš(ао|о|'о)**

Неко би користио запис **(дош(ао|о|'о))|(doš(ао|о|'о))** али додатне заграде на крајевима записа дописивања, иако јасније говоре о чему се ради, нису потребне јер **дописивање** (означено наглашеним фонтом црне боје) има већи приоритет од спољашње **алтернације** (означене наглашеним фонтом зелене боје).

Комбиновање уније (алтернације) и дописивања (2)

- ▶ Претпоставимо да у тексту треба пронаћи све имејл адресе које припадају неком од домена google, yahoo, hotmail, на пример:
 - ▶ mika@google.com,
 - ▶ pera@yahoo.com,
 - ▶ sima123@hotmail.com.
- ▶ Ако занемаримо произвољно корисничко име и завршетак домена (.com), обавезни заједнички део свих наведених адреса је карактер @ за којим следи назив неког од тражених домена. Одговарајући скраћени регуларни израз би био

@(google|yahoo|hotmail)

Комбиновање уније (алтернације) и дописивања (3)

- ▶ Изградимо поступно регуларни израз за опис поздрава:
 - ▶ Добро јутро
 - ▶ Добро вече
 - ▶ Добар дан
- ▶ Први корак: обична, двострука алтернација (унија)
Добро јутро|Добро вече|Добар дан
- ▶ Други корак: извлачимо заједнички префикс у првој алтернацији, па је цео израз сада само једна (спољашња) алтернација (зелена боја):
Добро (јутро|вече)|Добар дан
- ▶ Трећи корак: поново извлачимо заједнички префикс за изразе повезане спољашњом алтернацијом (зелена боја)
Доб(ро (јутро|вече)|ар дан)

Џокер знак

- ▶ Претпоставимо да желимо у тексту да пронађемо све речи које почињу са **ме**, завршавају се са **ак**, а између се налази било које слово.
- ▶ Једно приближно решење би користило тачку као џокер знак, тј. метакарактер који мења било који карактер (сем знака за нови ред):

ме.ак

- ▶ Проблем са овим решењем је што би у тексту

Сима има **мерак** да испали **метак**. Погоди **ме ако** смеш.

препознао и оне делове који нису једна реч већ две речи раздвојене размаком (**ме ако**).

- ▶ Непрецизност препознавања приликом употребе џокер знака је разлог што се он ретко користи, а уместо њега се примењују карактерске класе.

Карактерске класе као специјални џокер знаци

- ▶ У случају када регуларни израз треба да означи један произвољан симбол неког подскупа азбуке, Notepad++ нуди другачији начин означавања — карактерске класе.
- ▶ Подскуп азбуке који описује карактерска класа може да се зада на неколико начина. Основни начин подразумева набрајања свих карактера подскупа између средњих заграда, [и].
- ▶ На пример, произвољан вокал у српском језику, тј. скуп {а,е,и,о,у}, може да се опише на два начина:
 - ▶ операцијом алтернације (уније) а|е|и|о|у
 - ▶ карактерском класом [аеиоу]
- ▶ На први поглед се ништа не добија заменом алтернације карактерском класом.

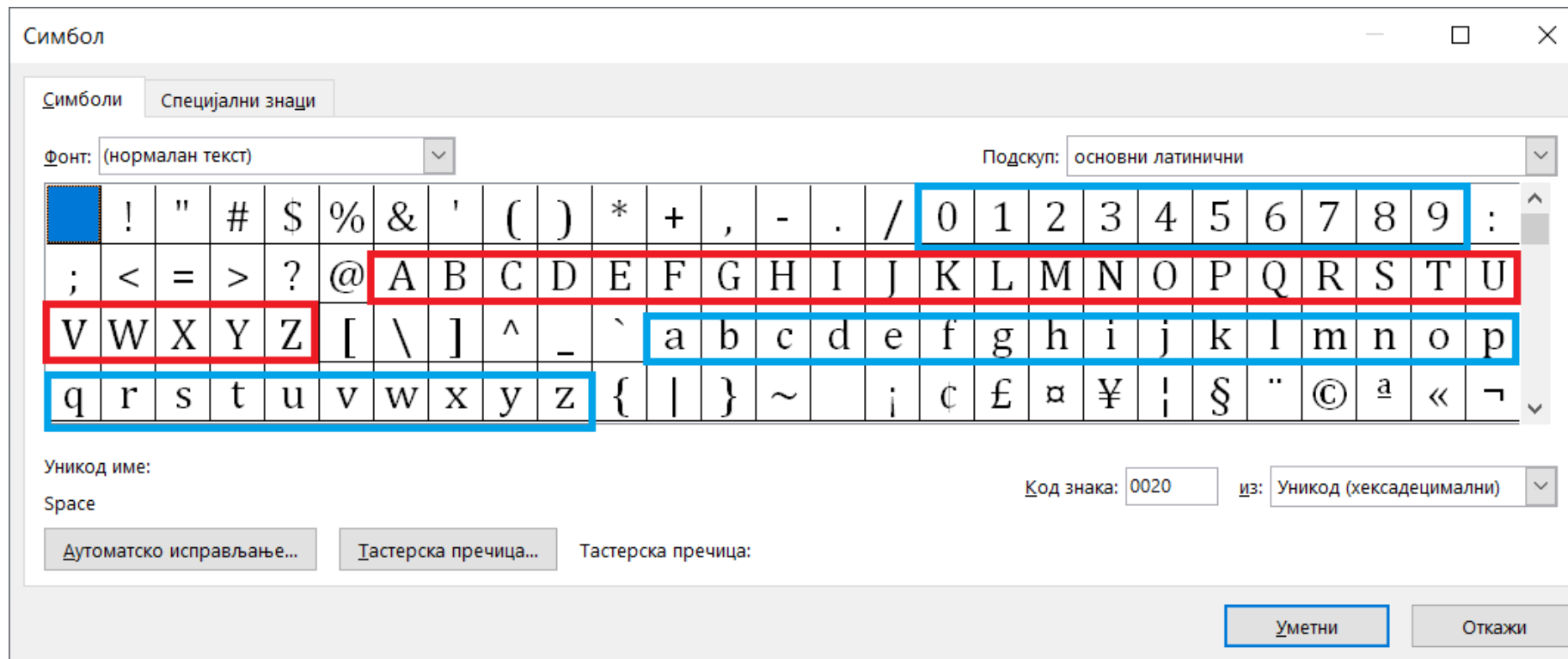
Карактерске класе као џокер знаци за опсег карактера

- ▶ Једна од погодности код коришћења карактерских класа је могућност да се скраћеним изразом опише подскуп свих карактера Unicode-а који се налазе између два задата карактера и на тај начин дефинише специјалан џокер знак који мења било који елемент тог подгрупа.
- ▶ На пример, уместо коришћења алтернатије и набрајања свих 26 великих слова енглеске абецеде

A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

довољно је у средњим заградама, [и], навести само прво и последње слово и цртицу између њих (**A-Z**) као ознаку да је у питању опсег. Добијени израз **[A-Z]** функционише као џокер за произвољно велико слово енглеске абецеде само зато што су енглеска слова исте величине представљена у Unicode-у **на суседним позицијама и у истом поретку као у абецеди**. Слично важи и за декадне цифре.

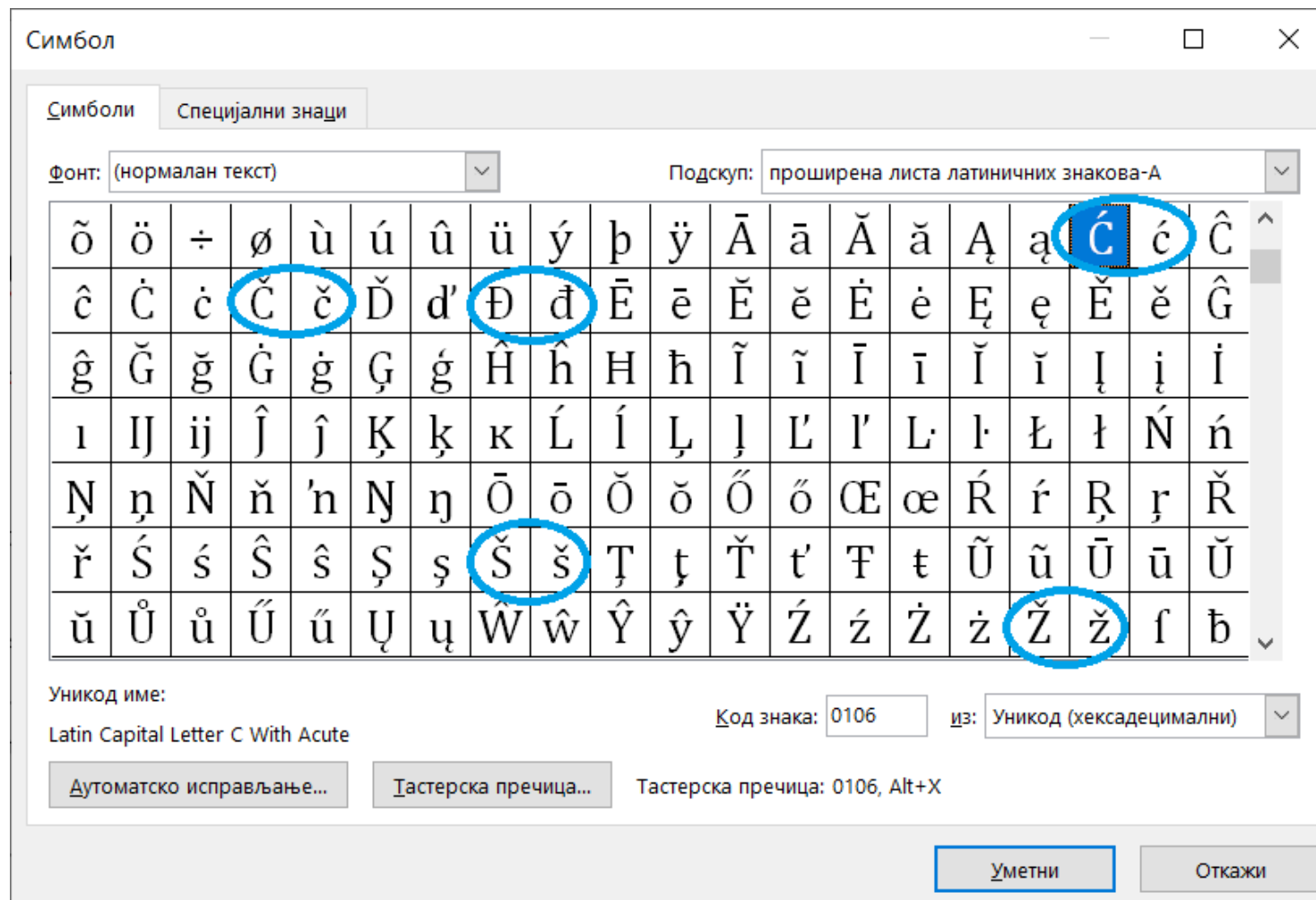
Распоред слова енглеске абецеде (Unicode, MS Word, дијалог Insert Symbol)



Често коришћене елементарне карактерске класе

- ▶ Произвољна декадна цифра **[0-9]** или **\d**
- ▶ Произвољно велико слово енглеске абецеде **[A-Z]**
- ▶ Произвољно мало слово енглеске абецеде **[a-z]**
- ▶ Произвољно слово енглеске абецеде **[A-Za-z]**
- ▶ Нажалост, наша слова (ћирилична и латинична) се не могу обухватити прецизним формулама типа а-ш, односно а-ž, али постоје заобилазна решења.

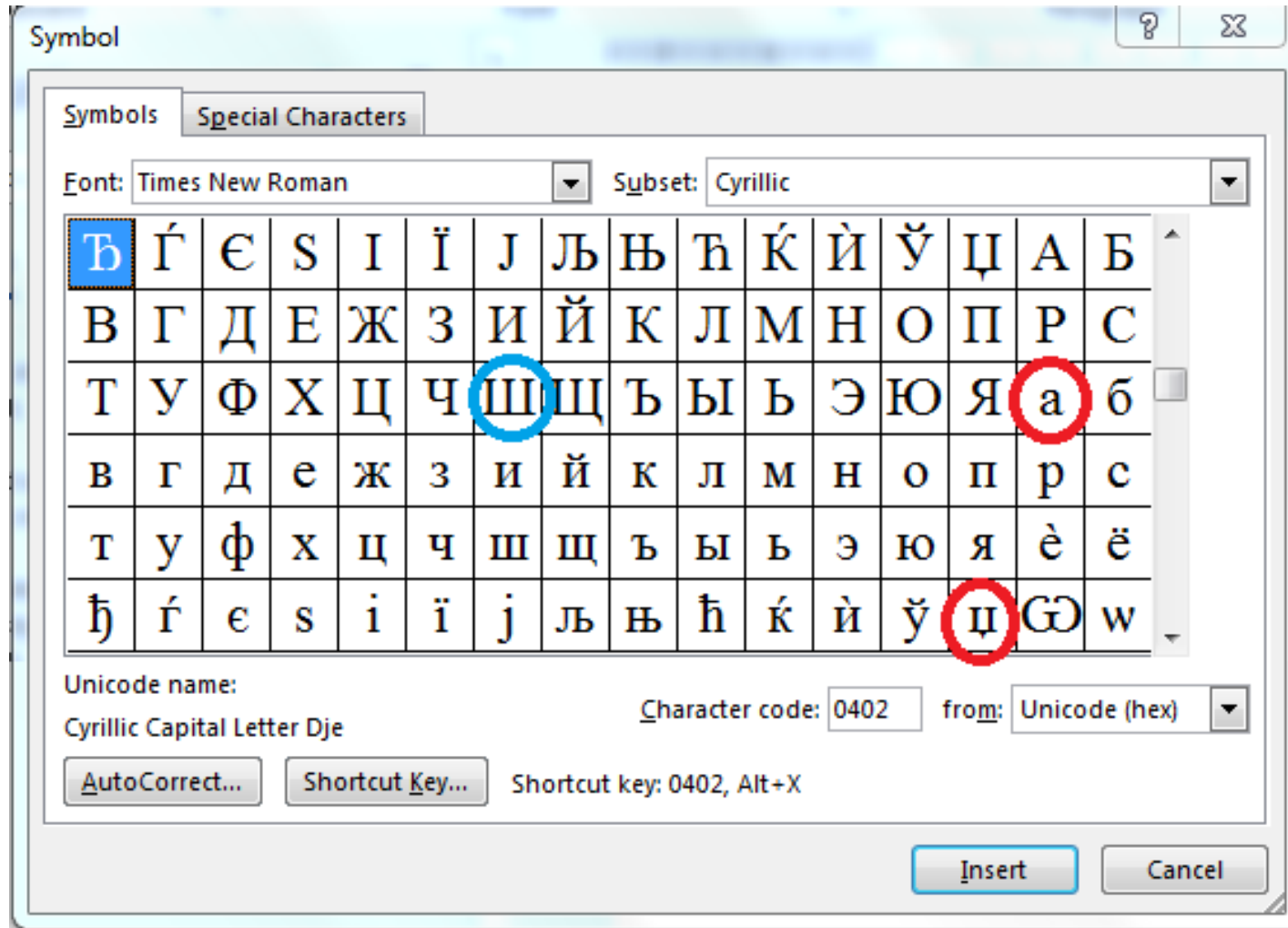
Распоред латиничних слова специфичних за српски језик (Unicode, MS Word)



Карактерске класе за слова српске латинице

- ▶ Произвољно велико слово српске латинице (и енглеског алфабета, укључујући Q, Y, W) **[A-ZĆČĐŠŽ]**. У овом примеру се комбинују набрајање карактера (**ĆČĐŠŽ**) и опсег (**A-Z**).
- ▶ Произвољно мало слово српске латинице **[a-zćčđšž]** (и енглеског алфабета, укључујући q, y, w). Такође се комбинују набрајање карактера (**ćčđšž**) и опсег (**a-z**).
- ▶ Произвољно слово српске латинице (али и слова енглеског алфабета, укључујући Q, q, Y, y, W, w) **[A-Za-zĆćČčĐđŠšŽž]**. Овај пример показује да се у истој класи могу комбиновати два или више опсега (**A-Z** и **a-z**), као и набрајање карактера (**ĆćČčĐđŠšŽž**).

Распоред ћириличних слова (Unicode, MS Word, дијалог Insert Symbol)



Карактерске класе за слова српске ћирилице

- ▶ Произвољно велико ћирилично слово српске азбуке (и понеко које није, дакле, апроксимација) [Ђ-Ш]
- ▶ Произвољно мало ћирилично слово српске азбуке (и понеко које није, апроксимација) [а-џ]
- ▶ Произвољно ћирилично слово српске азбуке (и понеко које није, апроксимација) [Ђ-Ша-џ]
- ▶ Опрез!! Цртица за опсег има специјално значење **само унутар** средњих заграда, па се разликују рег. изрази:
 - ▶ 0-9 препознаје скуп који садржи само једну ниску: 0-9 (карактери нула, цртица и девет);
 - ▶ [0-9] препознаје скуп ниски {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, тј. било коју декадну цифру.

Негација карактерске класе

- ▶ Понекад је лакше задати шта не желимо да препознамо. На пример, ако претражујемо ниску за коју знамо да се састоји само од ћириличних слова и желимо да пронађемо само појединачне сугласнике, тада, уместо да набрајамо њих 25, можемо захтевати „било који карактер који није вокал“ (пошто се ниска састоји само од слова, то се у овом случају сведе на тражење сугласника).
- ▶ Симбол **^** („капица“) непосредно после отворене средње заграде (**[^**) указује да уместо скупа карактера наведеног у класи заправо тражимо његов комплемент, тј. било који карактер (сем знакова за нови ред) који није наведен у класи. Тако регуларан израз **[^аеиоу]** представља један произвољан карактер који није ћирилични вокал.
- ▶ Са негацијом треба бити опрезан јер она не говори шта желимо већ шта не желимо да препознамо, што понекад може превише да сузи или да прошири скуп резултата. Тако претходни регуларни израз не препознаје само сугласнике, већ и све остале карактере који нису вокали!

Опрез! Честе почетничке грешке

- ▶ Цокери и карактерске класе увек представљају један једини карактер и могу се увек заменити алтернацијом.
- ▶ Обратно, алтернација **карактера (!)** се може заменити цокером или карактерском класом, али не и алтернација ниски!
- ▶ Према томе, регуларни изрази $(d|n)$ и $[dn]$ имају исто значење (карактер d или карактер n), али регуларни изрази $(d|n)$ и $[d|n]$ немају (овај последњи значи карактер d или карактер $|$ или карактер n).
- ▶ Још тежа грешка: $(da|ne)$ и $[da|ne]$ нису исти! Први препознаје ниске da и ne , док други препознаје један карактер који може бити d , a , $|$, n или e .

Опрез! Честе почетничке грешке (2)

- ▶ Како би изгледао регуларни израз за редни број дана у месецу, тј. цео број од 1 до 31 (тима обухватамо и месеце са мањим бројем дана)? Чињеница да је у питању опсег (интервал) природних бројева погрешно сугерише да треба применити карактерску класу са опсегом

[1-31]

- ▶ Погрешно! Претходни регуларни израз представља један карактер и то или један карактер из опсега 1-3 или карактер 1; другим речима, један од карактера 1, 2, 3.
- ▶ Прави узрок грешке је што вредности редног броја дана у месецу (1-31) не треба третирати као бројеве, већ као ниске (низове карактера), јер регуларни изрази описују ниске неког скупа користећи њихове заједничке особине.

Регуларни израз за редни број дана у месецу

- ▶ Овај пример илуструје како је понекад погодније разбити скуп на мање, правилне делове и посебно их описати, а онда применити алтернацију.
- ▶ Једноцифрени редни број дана у месецу се описује као $[1-9]$.
- ▶ Бројеви 10–19 су двоцифрени и прва цифра је 1, док друга може бити произвољна декадна цифра. Слично, бројеви 20–29 су двоцифрени, прва цифра је 2, док друга може бити произвољна декадна цифра. Дакле, бројеви 10–29 су двоцифрени и прва цифра је или 1 или 2, док друга може бити произвољна декадна цифра. Дакле, број из опсега 10–29 можемо описати изразом $[12][0-9]$. То је краћи запис у односу на

$1[0-9]|2[0-9]$

што значи да по потреби можемо извлачити не само заједнички префикс ниски, већ и заједнички суфикс (у овом примеру регуларни израз за другу цифру).

- ▶ Поучени претходним примером, израз $3[01]$ препознаје бројеве 30 и 31.
- ▶ Према томе, регуларни израз за редни број дана у месецу је унија претходних

$[1-9]|12[0-9]|3[01]$

Метакарактери квантификације (одређивања количине, тј. понављања)

- ▶ * (0 или више појављивања одређеног израза)
Регуларни израз (РЕ) **jaо*** одговара нискама **ја**, **јао**, **јаоо**, **јаооо**, итд. (карактери **ј**, **а** и коначан број слова **о**, укључујући и нула слова **о**).
- ▶ + (1 или више појављивања одређеног израза)
РЕ **jaо+** препознаје ниске **јао**, **јаоо**, **јаооо**, итд. (карактери **ј** **а** и коначан број слова **о**, али произвољно велик).
- ▶ ? (необавезно појављивање, једном или ниједном)
РЕ **јелени?** препознаје ниске **јелен** и **јелени**.
- ▶ {*n*} (појављивање тачно *n* пута, где је *n* природан број)
РЕ **w{3}** одговара ниски **www**.
- ▶ {*m*, *n*} (појављивање најмање *m* пута и највише *n* пута, где су *n*, *m* природни бројеви).
РЕ **-{1,3}** одговара нискама **-**, **--**, **---** (једна до три цртице).

Комбиновање алтернације и понављања

- ▶ Квантификација или понављање не морају да се односе само на појединачне карактере, већ и на ниске и саме регуларне изразе, најчешће дописивања, алтернације, цокере и карактерске класе.
- ▶ Претпоставимо да анкетирамо 6 особа које навијају за Звезду или Партизан и чије одговоре на питање за кога навијају редом бележимо користећи размак као сепаратор (на пример, могући низ одговора би био *Звезда Партизан Партизан Звезда Звезда Партизан*). Ако ради боље уочљивости размак у регуларним изразима посебно нагласимо ознаком `_`, тада би приближни регуларни израз за опис ових одговора био `(Звезда_|Партизан_){6}`. Приближни је зато што препознаје један размак вишка иза последњег одговора. Тачни одговори би били
 - ▶ `(Звезда_|Партизан_){5}(Звезда|Партизан)`
 - ▶ `(Звезда|Партизан)(_Звезда|_Партизан){5}`
- ▶ Овај пример илуструје да се у општем случају не понавља један конкретан одговор (дакле, није нужно *Звезда* 6 пута одговор, нити *Партизан* 6 пута одговор), већ се 6 пута понавља могућност да одговор буде или *Звезда* или *Партизан*, при чему сваки конкретан одговор не зависи од осталих.

Понављање цокера и карактерских класа (1)

- ▶ `.{3}` препознаје низ од три произвољна карактера који не морају бити истоветни.
- ▶ Природни број (укључујући и нулу) могао би да се препозна регуларним изразом `[0-9]+`, односно `\d+`. У овом примеру не треба тумачити регуларни израз (само) као понављање једне конкретне декадне цифре (на пример, 111, 99, 8888 итд.), већ израз треба читати као „једна или више декадних цифара“, при чему на свакој позицији може бити произвољна декадна цифра.
- ▶ Број индекса на Филолошком факултету се састоји од 4 цифре године уписа и 5 цифара редног броја при упису (укупно 9 цифара), што може да се опише регуларним изразом `[0-9]{9}`, односно `\d{9}`.

Понављање цокера (2)

- ▶ На сличан начин, реч природног језика се може грубо апроксимирати регуларним изразима:
 - ▶ **[A-Za-z]+** (за енглески текст: низ слова енглеске абецеде),
 - ▶ **[Ђ-Ша-џ]+** (за српски текст на ћирилици: низ слова српске ћирилице),
 - ▶ **[A-Za-zŠĐČĆŽšđčćž]+** (за српски текст на латиници: низ слова српске латинице).
 - ▶ **[A-Za-zŠĐČĆŽšđčćž]+[Ђ-Ша-џ]+** (за српски текст у коме се појављује и ћирилица и латиница: низ слова на ћирилици или низ слова на латиници).
- ▶ У последњем примеру имамо комбиновање карактерских класа, квантификатора (понављања) и алтернације (намерно истакнуте зеленом бојом). Карактерска класа има већи приоритет од понављања (квантификације), а алтернација (унија) најмањи приоритет, зато заграде нису потребне.

Комбиновање карактерских класа, дописивања и квантификације

- Проблем са регуларним изразом $[0-9]^+$ за природни број је што препознаје и записе бројева у којима се на првој позицији налази бар једна нула (0000, 0023, 00000378 итд.).
- Природан број се може описати као (бар) једна цифра различита од нуле за којом може, али не мора да следи једна или више произвољних цифара. Дакле, природан број се може поделити на:
 - прву цифру (различиту од нуле) којој одговара регуларни израз $[1-9]$ и
 - преостали део записа броја који може бити празан или се састојати од једне или више цифара, што записујемо $[0-9]^*$.

Ознака $*$ се назива Клинијева звездица или Клинијево затворење и означава да се РЕ испред ње не мора појавити или се може појавити једном или више пута („понављање нула или више пута“)

Дописивањем ова два РЕ добијамо РЕ за природан број:

$[1-9][0-9]^*$

Комбиновање негације карактерских класа, дописивања и квантифікације

- ▶ Негација карактерских класа је погодна за издвајање делова текста који су неким граничником (енгл. *delimiter*) или паром граничника одвојени од остатка текста (типичан пример су цитати под наводницима).
- ▶ Претпоставимо да је потребно у HTML-документу пронаћи и обрисати све етикете. Знајући да су етикете текст између угластих заграда, **<** и **>** (на пример, `<html>`, `<head>`, `<body>`, `<h1>`, `<p>`, `<ul>`, `<li>`, `<table>` итд.), регуларни израз за препознавање етикете би изгледао овако:

`<[^>]+>`

Комбиновање негације карактерских класа, дописивања и квантификације (2)

- ▶ У оквиру регуларног изрази `<[^>]+>` могу се уочити следећи структурни делови који се редом дописују:
 - ▶ Карактер `<` који представља самог себе, односно означава почетак HTML етикете;
 - ▶ Карактерска класа `[^>]` праћена знаком квантификације `+` чиме је означен произвољан текст између угластих заграда, `<` и `>`, тј. низ од једног или више било каквих карактера (`+`) од којих ниједан није карактер `>` (`[^>]`);
 - ▶ Карактер `>` који представља самог себе, односно означава крај HTML етикете.
- ▶ Овај пример показује моћ негације карактерске класе комбинован са квантификацијом, којима је описан произвољан назив етикете (отворене и затворене) без потребе да се сва та имена посебно наводе.

Контекстни метакарактери (сидра)

- ▶ Приликом претраживања текста регуларним изразима веома је важно знати шта је контекст претраге. У општем случају контекст претраге је ниска, што може бити текстуална реч, текстуална линија или чак цео текст. Сидра се користе за означавање почетка (^), односно краја (\$) контекста претраге. Notepad++ их користи да означи **почетак и крај текстуалне линије**, али то не значи да ће резултат претраге бити ниска која цела припада једној линији.
- ▶ XML Schema претпоставља да се регуларни израз увек односи на комплетну ниску која представља садржај неког XML елемента или атрибута, па не користи сидра ^ и \$ јер их увек подразумева.

Сидра (Notepad++ и XML Schema)

► Notepad++

1. **^[a-ц]** проналази мало ћирилично слово на почетку **линије**;
2. **[a-ц]\$** проналази мало ћирилично слово на крају **линије**;
3. **[a-ц]** проналази мало ћирилично слово било где у **линији**;
4. **^[a-ц]\$** проналази **линије** које садрже тачно једно мало ћирилично слово („између почетка и краја линије је тачно једно мало ћирилично слово“).

- XML Schema користи само 3. облик регуларног израза, **[a-ц]**, али са значењем сличним 4. регуларном изразу: **садржај XML елемента (атрибута)** је тачно једно мало ћирилично слово („између почетка и краја XML елемента (атрибута) је тачно једно мало ћирилично слово“).

Сидра (програмски језици Python, JavaScript, PHP)

1. **^[a-ц]** проверава да ли се на почетку задате **ниске** налази мало ћирилично слово;
2. **[a-ц]\$** проверава да ли се на крају задате **ниске** налази мало ћирилично слово;
3. **[a-ц]** проверава да ли се било где у задатој **ниски** налази мало ћирилично слово;
4. **^[a-ц]\$** проверава да ли је садржај задате **ниске** тачно једно мало ћирилично слово („између почетка и краја ниске је тачно једно мало ћирилично слово“).

Пазите на различита значења „капице“

- ▶ Значење „капице“ зависи од контекста и имплементације:
 - ▶ `^[аеиоу]`
 - ▶ вокал на почетку линије (Notepad++), односно ниске (Python, JavaScript). Овај регуларни израз је неприменљив кад је у питању XML Schema јер се `^` не користи као сидро већ искључиво у карактерским класама као негација.
 - ▶ `[^аеиоу]`
 - ▶ карактер који није вокал (у свим горе наведеним имплементацијама).
 - ▶ `^[^аеиоу]`
 - ▶ карактер на почетку линије (Notepad++), односно ниске (Python, JavaScript) који није вокал. Овај регуларни израз је неприменљив кад је у питању XML Schema јер се `^` не користи као сидро већ искључиво у карактерским класама као негација.

„Врачара“

- ▶ Посебан метакарактер је „врачара“, тј. обрнута коса црта (`\`). Њена улога је двострука:
 - ▶ „укида“ специјално значење другим карактерима
 - ▶ `\.` је само тачка;
 - ▶ `\\` је само вертикална црта;
 - ▶ `\( \) \[ \] \{ \}` су само заграде (редом обичне, средње и витичасте);
 - ▶ `\* \+ \?` редом представљају звездицу, плус и упитник;
 - ▶ `\^` је само „капица“, `\$` је само знак за долар;
 - ▶ `\\` представља обичну обрнуту косу црту.
 - ▶ додељује специјално значење појединим карактерима на пример, карактерима `t`, `r`, `n` који се користе за означавање белина.

Белине: размак, табулатор, знак за нови ред

- Размак у регуларним изразима се означава на уобичајени начин, али није свеједно када се наводи. Размак у регуларним изразима треба наводити **искључиво ако га је потребно препознати у тексту**. У противном, корисников регуларни израз мења значење и вероватно неће наћи оно што се тражи. Ради боље уочљивости, размак у регуларним изразима ћемо посебно наглашавати ознаком `␣`.
- Табулатор се препознаје регуларним изразом `\t`.
- Знаци за крај реда CR (енгл. carriage return) и LF (енгл. line feed) се редом означавају регуларним изразима `\r` и `\n`. Према томе, крај реда можемо препознати регуларним изразима:
`\r\n` (Windows), `\n` (Linux), односно `\r` (Mac).

Скраћенице за карактерске класе

- ▶ У неким имплементацијама (укључујући Notepad++) врачара појединим карактерима (пored већ виђених **t**, **r**, **n**) додељује специјално значење те се они могу користити као скраћенице за неке чешће коришћене карактерске класе:
 - ▶ **\s** се користи као скраћеница за произвољну белину, тј. **[\t\r\n]**, док је **\S** скраћеница за негацију произвољне белине, тј. **[^ \t\r\n]** (у карактерским класама изостављамо пар белина које су обухваћене изразом **\s**, тзв. form feed и вертикални табулатор, јер се данас ретко користе).
 - ▶ **\d** се користи као скраћеница за **[0-9]**, а **\D** као скраћеница за **[^0-9]**.

Опрезно са скраћеницама карактерских класа!

- ▶ `\w` се користи као скраћеница за било који Unicode карактер који у неким програмским језицима може бити део идентификатора (тј. део имена променљиве или функције). Нажалост, значење `\w` није исто код свих програмских језика пошто користе различита правила за имена идентификатора. Тако у случају програма Notepad++ и програмског језика Python `\w` означава:
 - ▶ велико или мало слово неког природног језика (укључујући енглеску абегеду, српску ћирилицу и латиницу, али и азбуке разних других европских и светских језика),
 - ▶ декадну цифру 0–9 или
 - ▶ подвлаку или доњу црту `_`.
- ▶ Међутим, у неким другим програмским језицима (на пример, JavaScript-у), `\w` има сужено значење, тј. користи се као скраћеница за `[a-zA-Z0-9_]`, тј. од слова која препознаје у обзир долазе само слова енглеске абегеде. Стога класу `\w` треба користити са опрезом и углавном ћемо је избегавати.
- ▶ `\W` се користи као негација карактерске класе `\w`.

Врачара уклања специјално значење метакарактера

- ▶ Подсетимо се задатка проналажења свих имејл адреса у тексту, које припадају неком од домена google, yahoo, hotmail, на пример:
 - ▶ mika@google.com,
 - ▶ pera@yahoo.com,
 - ▶ sima123@hotmail.com.
- ▶ Ако занемаримо само произвољно корисничко име, обавезни заједнички део свих наведених адреса је карактер @ за којим следи назив неког од тражених домена (укључујући и завршетак .com). Међутим, како је тачка метакарактер (цокер који мења било који карактер), користимо врачару:

@(google|yahoo|hotmail)\.com

Претрага регуларним изразима у програму Notepad++

- Пронађимо у тексту све датуме структуре гггг-мм-дд (д = дан, м = месец, г = година), на пример 2019-10-31.
- Најгрубља апроксимација оваквог датума је „четири цифре, цртица, две цифре, цртица, две цифре“, односно

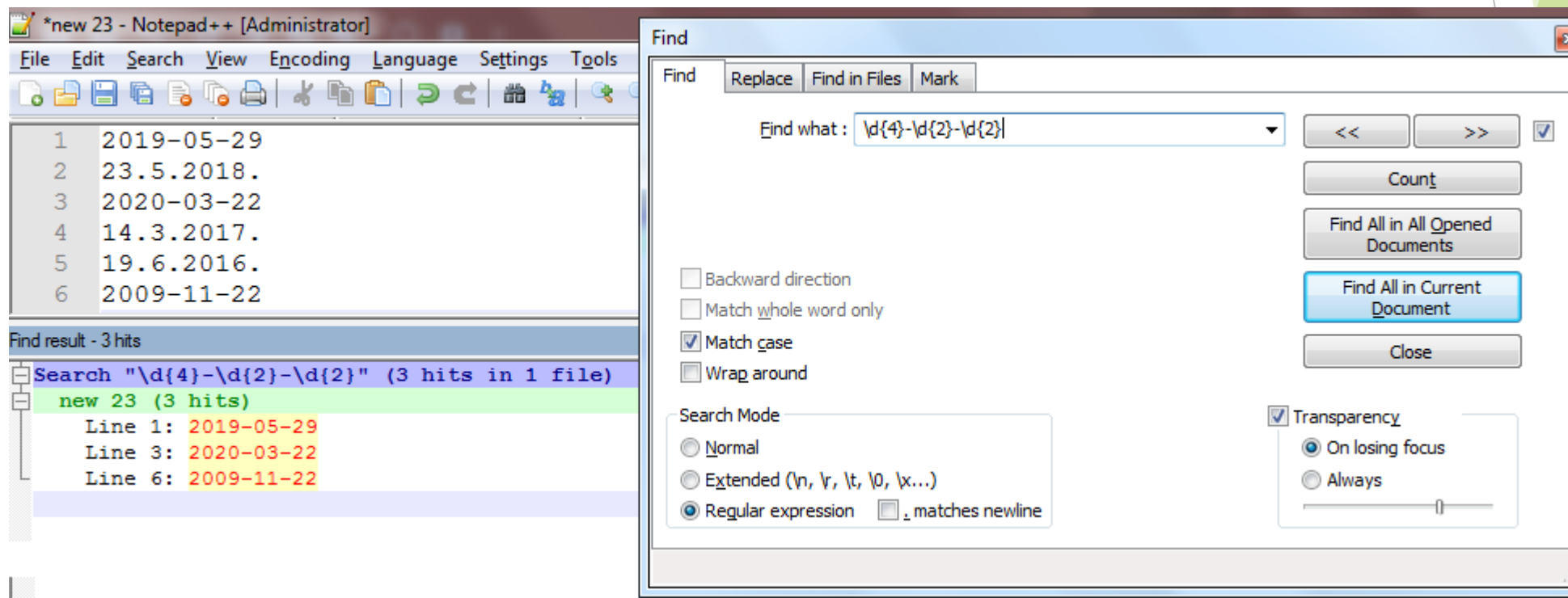
Find: `[0-9]{4}-[0-9]{2}-[0-9]{2}`

или

Find: `\d{4}-\d{2}-\d{2}`

Анализа резултата претраге

- Понекад је потребно пажљиво анализирати колико је употребљени РЕ прецизан, поготово када се изводе операције замене. У ту сврху је корисно дугме *Find All in Current Document* које у посебном прозору приказује све линије (са редним бројем линије) које садрже тражени образац, као и укупан број резултата. Ако нас занима само укупан број резултата које би образац пронашао, довољно је искористити дугме *Count*. Обе опције се могу користити и случају када се користи „нормална“ претрага без регуларних израза.



Пример: конверзија датума у тексту помоћу регуларних израза и „магије заграда“

- Пример: заменити све датуме структуре гггг-мм-дд (д = дан, м = месец, г = година) одговарајућим датумима структуре дд.мм.гггг у једном потезу. На пример, на овај начин се у тексту датум облика 2019-10-31 замењује датумом облика 31.10.2019.

Find: ([0-9]{4})-([0-9]{2})-([0-9]{2}) или Find: (\d{4})-(\d{2})-(\d{2})

Replace: \3.\2.\1.

- Коришћење обичних заграда у регуларним изразима служи не само за избегавање, односно наметање приоритета, већ и за „памћење“ делова препознатог обрасца. Сваком пару заграда у регуларном изразу одговара једна промењива (највише њих 9) чији бројеви у ознакама \1, \2,... \9 представљају редни број одговарајућег пара заграда посматрано слева надесно. Део структуре који је препознат у оквиру неког пара заграда, меморише се у одговарајућој промењивој чија вредност може да се користи приликом замене текста (у овом примеру \1 = година, \2 = месец, \3 = дан). Ово својство регуларних израза приликом замене текста се још назива „магија заграда“.

Анотација регуларним изразима

Регуларне изразе се могу користити при изради семинарског рада, тј. при обележавању пагинације (претпоставка је да је редни број странице у књизи у посебној линији и да та линија не садржи ништа више):

Find: `^([0-9]+)$` или Find: `^(\\d+)$`

Replace: `<pb n='1' />`

као и приликом обележавања пасуса (у случају да свака линија текста у датотеци увек представља један пасус, тј. крај линије је и крај пасуса):

Find: `^(\\.+)$`

Replace: `<p>\\1</p>`

- Претходни израз може још да се прецизира, али о томе други пут. Више о регуларним изразима учићете у наредним курсевима.

Приоритет метакарактера и завршни преглед регуларних изрази

- ▶ Опадајући приоритет метакарактера:
 - ▶ Заграде (и) имају највећи приоритет
 - ▶ Карактерске класе [и]
 - ▶ Метакарактери квантификације * + ?
 - ▶ Дописивање и цокер знак .
 - ▶ Контекстни метакарактери ^ \$
 - ▶ Алтернација | има најмањи приоритет
- ▶ Следећи слајд даје преглед метакарактера које користе РЕ у програму Notepad++ и примере њихове употребе.



Метакарактер	Чита се	Значење	Пример	Препознаје
	или	Унија (алтернација) ниски	као ко к'о	трочлани скуп ниски {као, ко, к'о}
()	заграде	Груписање	к(ао о 'о)	трочлани скуп ниски {као, ко, к'о}
^	капица	Почетак линије	^Б	све линије које почињу словом Б
\$	долар	Крај линије	сти\$	све линије које се завршавају на сти
.	тачка	Џокер, ма који карактер	^..\$	све линије са тачно 2 карактера
[<i>каракџери</i>]	класа	Карактерска класа (унија појединачних карактера)	[аеиоу]	један произвољан вокал, тј. елемент петочланог скупа {а, е, и, о, у}
[^ <i>каракџери</i>]	негација класе	Негација карактерске класе	[^аеиоу]	један произвољан карактер који није вокал, тј. не припада скупу {а, е, и, о, у}
*	звездика	0 или више појављивања	мјауу*	{мјау, мјауу, мјаууу, мјауууу,...}
+	плус	1 или више појављивања	мјау+	{мјау, мјауу, мјаууу, мјауууу,...}
?	упитник	Необавезно појављивање (једном или ниједном)	ка?о	двочлани скуп ниски {као, ко}
{ <i>m, n</i> }	понав- љање	Понављање најмање <i>m</i> пута, а највише <i>n</i> пута	(0 1){1,3}	скуп највише троцифрених бинарних бројева {0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111}
{ <i>n</i> }	понав-	Понављање тачно <i>n</i> пута	(0 1){3}	скуп троцифрених бинарних бројева