**Regular Expressions**

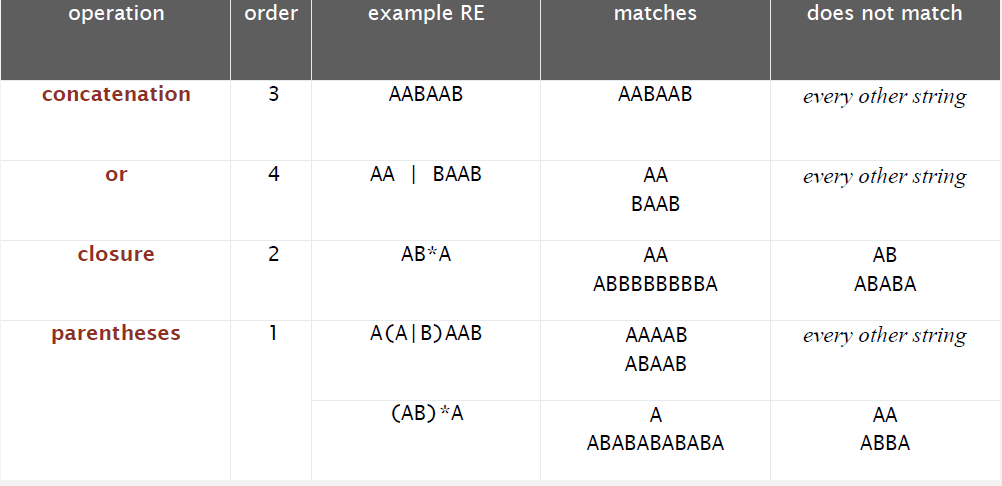
REGEX na mundësojnë përfshirjen e disa stringjeve me një string të vetëm, me anë të kti stringu ne mund të bëjm kërkim të disa stringjeve tjera gjë që e shpejton kërkimin dhe me një maqim të vetme mund të kërkojm disa stringje.

**SubString Search :** Gjen një string në tekst.

**Pattern matching :** Gjej një nga bashkësitë specifike të stringjeve në tekst.

Pattern matching aplikimet : Virus sakneret, filtrime të tekstit, validim i formularëve etj.

Një **regular expression** është një symbol që specifikon një bashkësi të stringjeve.

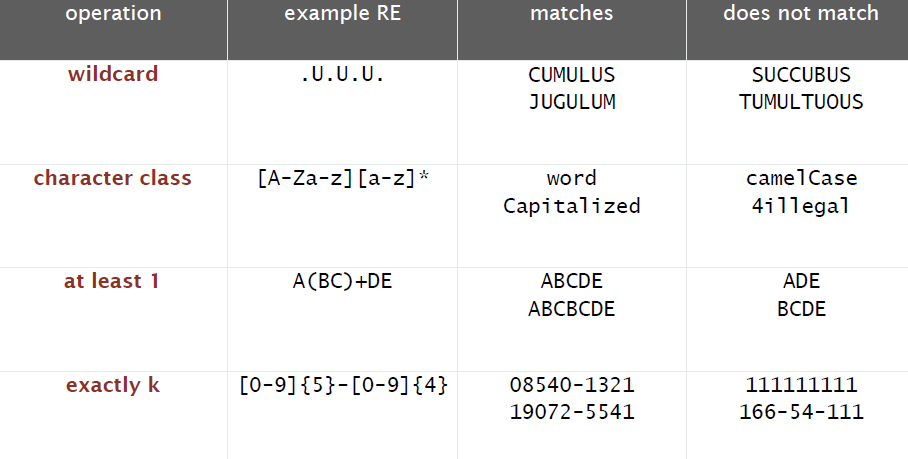


**Concatenation** : Bën match vetëm karakteret që janë ekzakt të njëjta, me ato në regex.

**OR :** Bën match karakterin e majt ose të djathtë, por vetëm njërin në të njëjtën kohë ( **|** ).

**Closure :** Përdor simbolin **\*** , karakteri që vendoset para kti ylli do ket të drejt të shfaqet disa herë deri në infinit gjë që na jem . infinit mundësi për match , ose mund të mos shfaqet asnjë herë ( **\*** ).

**Parentheses :** Ka rolin e njëjtë si kllapat në matematik na mundson ti bëjm bashk dy a më tepër karaktere **( )**.

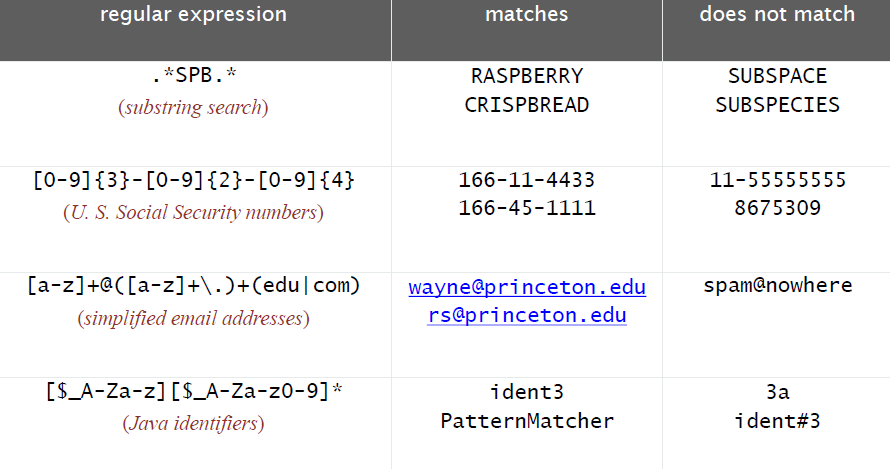


**Wildcard :** Me wildcard ne mund të zëvendësojmë cilindo karakter që ekziston, e shenojmë me ( **.** )

**Character calss :** Na mundëson të marrin një klass të karaktereve , për të klasifikuar klasa të karaktereve përdorim dy kllapat e mesme **[ ]** , gjithashtu edhe **–** për të caktuar një range nga karakteri në karkater.

**At least 1 :** Karakteri që vendoset para simbolit të at least 1 , do ket të drejt të shfaqet më së paku një her ose disa her , deri në infinit gjë që na jep infinit mundësi për match ( **+** ). Na mundpson edhe shkurtimin e regex.

**Exactly k :** Përcakton se sa her duhet të shfaqet karkateri që gjindet para **{N}** , kllapat gjarprore dhe në mes numri sa her duam të shfaqet karkteri.



**SubString search**  me regex është kërkimi që përdoret më së shumti në regex , spese me pak rreshta definon shumë. Shenohet me **.\*Char.\*** , pra wildcard clouser karakteri wildcard clouser.

Regex janë të leta për tu shkruar , por shumë të veështira të lexohen, shumica e gjërave në pc ruhen si regex.

Pattern match implemetimi mund të bëhet në dy mënyra : me DFA ose e NFA, DFA na garanton koë lineare dhe nuk ka backup , mirpo për një RE do duhej të ndërtoheshin disa DFA, NFA na siguron koh kuadratike dhe nuk ka back up mirpo ska nevoj që për një RE të ndërtohen shum maqina.

Për të përcaktaur nëse një string ka match në automat ,DFA e ka më let sepse ka vetëm një transicion, ndërsa NFA mund të ket disa transicion të cialat duhet kontrolluar.

**NFA :** quhet nfa sepse ka më shumë se një rrug qp tp qonp në final state.

**Teorema e Kleene-it** : Për gjdo DFA ekziston një RE që përshkruan të njëjtën bashkësi të stringave, për gjdo RE ekziston një DFA që njeh të njëjtën bashkësi të stringave.

**NFA-REGEX-Rregullat-E-Konstruktimit**

**Clousers** \* : Bën një **ε** transicion.

**Karakter** 0-Z : Bën fillimisht një match transicion pastaj kontrollon a është clouser karakteri I radhës nëse po I lidh të dya midis veti.

**Kllapa hapëse (** : Bën një **ε** transicion dhe vendos indexin e kllapës në stack.

**OR |** : E Vendos indexin e OR në stack dhe vazhdon tek karakteri i radhës pa bërë ndonjë transicion.

**Kllapa mbyllëse )** : ka tre rregulla : **1.**Bën një **ε** transicion. **2.**Nxierr një index nga stack, nëse index I nxierrur I përket një kllape mbyllëse m atëher NFA ndalon së nxierruri elemente nga stacku dhe vazhdon në rregullën e radhës, por nëse nuk e gjen kllapën mbyllëse vazhdon të nxierr elemente nga stack deri të arrin ta nxierr një kllap hapëse, nëse gjat kti procesi ka nxierrur një index që I përket OR ( | ) , atëher indexin e kllapës hapse që në fund e kemi nxierr nga stacku e lidhim me një index më të madh se ai i OR ndërsa OR e lidhim me kllapën mbyllëse.**3.**Kontrollojmë nëse ka clouser \* nëse po kllapën hapse e lidhim me \* dhe anasjelltas.