Sistem za zaključivanje na bazi pravila u LegalRuleML jeziku

Nenad Gligorov\*, Nikola Ilić\*

\* Fakultet tehničkih nauka, Novi Sad, Srbija

[ngligorov@uns.ac.rs](mailto:ngligorov@uns.ac.rs), nikola\_ilic96@hotmail.com

*Apstrakt* — Predmet istraživanja ovog rada jeste predstava zakona upotrebom LegalRuleML jezika i donošenje zaključaka na osnovu pravila izvedenih iz LegalRuleML-a. Zamišljeno je da sistem koji će donositi zaključke bude deo sistema za takozvane autonomne automobile. Rešavanje ovog problema će doprineti tome da se nad određenim delovima zakona koje je moguće predstaviti u vidu pravila pomoću LegalRuleML jezika može rezonovati, odnosno odlučivati i zaključivati. Zakon predstavljen u LegalRuleML jeziku se parsira u defeasible logiku nad kojom se rezonuje pomoću SPINdle alata za rezonovanje. Na taj način se donose zaključci na osnovu pravila. Rezonuje se nad setom pravila predstavljenih u defeasible logici i određenim činjenicama na osnovu kojih rezoner donosi različite odluke za data pravila. Rezultati koji su dobijeni su od značaja za razvoj sistema za odlučivanje na osnovu pravila koji su izvedeni iz zakona. Pored toga od važnosti je i sam parser napisan za potrebe ovog rada koji može poslužiti za transformaciju LegalRuleML jezika u druge jezike defeasible logike nad kojim je moguće rezonovati.

*Ključne* *reči* — LegalRuleML, rule-based sistem, defeasible logika, parser

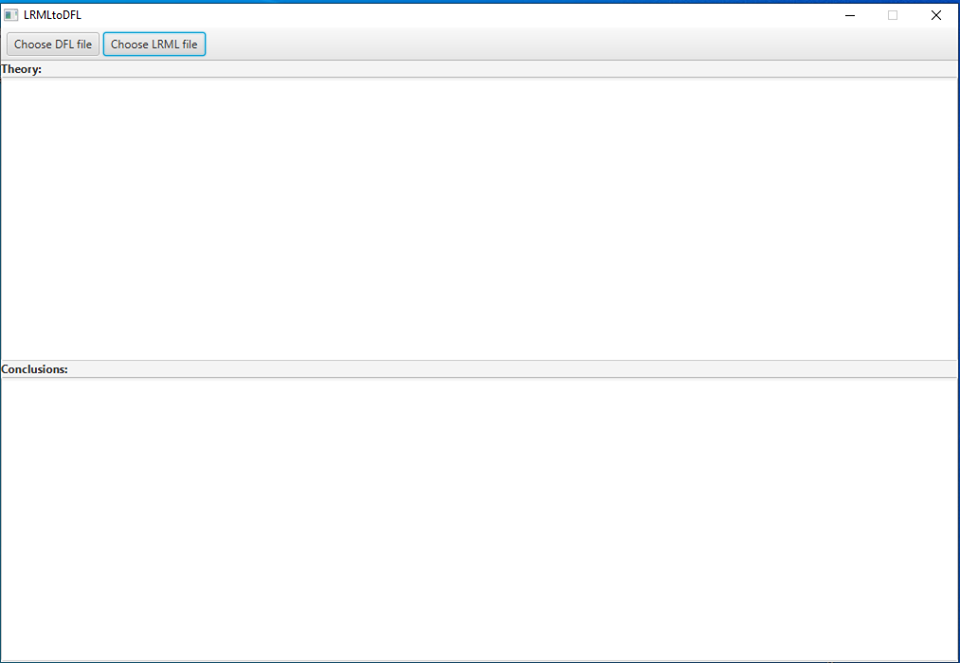
# Uvod

Ideja o autonomnim automobilima nije više samo stvar mašte, sada je i realnost. U vremenu kada su računari preuzeli primat u većini industrija i kada je njihova glavna uloga u automatizaciji određenih procesa tako su se razvijali i automobili koji određene procese mogu da obavljaju sami. Upozoravanje vozača u slučaju da zaspi za volanom, pomoć pri parkiranju, GPS, pa čak i automatsko upravljanje samo su neke od olakšica koje su omogućili računari.

Predmet ovog istraživanja su prethodno navedeni problemi, a cilj je razvijanje sistema koji će na osnovu određenih delova zakona o bezbednosti u saobraćaju donositi odluke pri određenim situacijama. U ovom radu to će se odnositi konkretno na prava prvenstva prolaza na raskrsnicama. Sistem je baziran na pravilima, odnosno na LegalRuleML [1] jeziku i *defeasible* logici [4] i SPINdle [3] rezoneru. Pri čemu je iskorišćen LegalRuleML za predstavu zakonskih odredbi u vidu pravila u formatu koji je razumljiv računaru. Pošto SPINdle ne može da rezonuje nad pravilima napisanim u LegalRuleML formatu bilo je potrebno napisati i parser koji će transformisati pravila iz LegalRuleML jezika u *defeasible* logiku nad kojom SPINdle može da rezonuje.

Rezultat ovog rada je rezonovanje nad izvedenim pravilima u *defeasible* logici. Ovo je od posebnog značaja za dalja istraživanja na polju automatskog rezonovanja, odnosno za donošenje odluka na osnovu pravila u LegalRuleML jeziku, ne samo zakona o bezbednosti u saobraćaju, već i drugih zakona koje je moguće predstaviti u vidu pravila u LegalRuleML jeziku.

Cilj je bio razviti sistem baziran na pravilima (eng. *rule-based system*) koji podržava rezonovanje na osnovu članova Zakona o saobraćaju u svojoj bazi znanja, kao i situacionih činjenica koje bi sistemu bile prosleđivane u realnom vremenu. Ovakav sistem bi mogao poslužiti kao jedna komponenta softvera autonomnih automobila, odnosno ona komponenta koja bi automobilu predočavala sve legalne akcije koje se mogu izvršiti u datoj situaciji.  
Rad je organizovan na sledeći način. U poglavlju 2. će biti reči o srodnim istraživanjima koja se odnose na ovaj rad. U poglavlju 3. je detaljnije opisan način na koji je predstavljen deo zakona upotrebom LegalRuleML jezika. U poglavlju 4. se govori o metodu koji je korišćen u ovom radu. U poglavlju 5. su predstavljeni rezultati rada. Poslednje poglavlje čini zaključak rada.



Slika – Prikaz korisničkog interfejsa sistema

# Srodna istraživanja

Jedno srodno istraživanje prikazano je u [3]*.* U radu je objašnjeno kako bi zakonske norme trebalo predstavljati pomoću LegalRuleML jezika kao i kako ta pravila zatim pravilno transformisati u format pogodan za rezonovanje. Takođe se govori i o implementaciji parsera koji bi transformisao *LegalRuleML* u sintaksu podržanu od SPINdle rezonera koji je takođe deo rada. Ipak, u poslednjoj javno dostupnoj verziji SPINdle rezonera, a to je verzija 2.2.4, jedini implementirani parseri su za XML i DFL jezik.

RuleML (*Rule Markup Language*) [2] predstavlja familiju jezika za predstavljanje, odnosno zapisivanje pravila na internetu. Osmišljen je za razmenu većine vrsta *Web* pravila u XML (*Extensible Markup Language*) formatu, dok je prvobitno specificiran u DTD-u (*Document Type Definition*). Korišćenje XML formata dovodi do uniformnosti između različitih platformi i jezika za zapisivanje pravila.

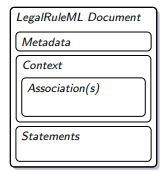
RuleML ima široku pokrivenost jezicima. Definisan je kao proširiva familija jezika, čiji modularni sistem XML šema dozvoljava razmenu pravila sa velikom preciznošću. Glavnu primenu našao je u Semantičkom *web*-u

LegalRuleML predstavlja specijalizaciju RuleML u domenu prava, proširuje ga podrškom za logiku

poništavanja, deontičku logiku, vremenske i prostorne aspekte pravila. Cilj LegalRuleML-a je premošćavanje razlika između zakona zapisanih prirodnim jezikom i semantičkih normi. Može se koristiti za modelovanje različitih zakona, pravila i regulativa.

Sam LegalRuleML dokument se sastoji iz tri glavne komponente, slika 2:

* metapodaci (eng. *metadata*)
* kontekst (eng. *context*)
* iskazi (eng. *statements*)



Slika 2 – Arhitektura LegalRuleML dokumenta

Metapodaci sadrže informacije o pravnim izvorima normi koje se modeluju u dokumentu, vremenske aspekte, kako pravnih izvora, tako i samog dokumenta, jurisdikcije u kojima su norme primenljive i detalje o stvaraocima pravne norme.

Sama apstraktna sintaksa LegalRuleML dokumenta predstavlja dokument kao bipartitni graf čiji su čvorovi instance klasa, a grane relacije između tih instanci. Konkretni primeri LegalRuleML sintakse su XML i RDF sintaksa. Za svrhe ovog rada koriščena je XML konkretna sintaksa.

LegalRuleML podržava modelovanje *defeasible* pravila unutar zakona. Ova osobina LegalRuleML-a je glavni razlog za odabir baš ovog formata zapisa zakona, jer umnogome olakšava kasnije prevođenje zakona u *defeasible* logiku.

Defeasible logika je jednostavan i efikasan formalizam baziran na pravilima. Kreirao ga je Donald Nute uz posebnu pažnju posvećenu efikasnosti i implementaciji.

Glavna intuicija ove logike je da dođe do verodostojnih zaključaka, polazeći od parcijalnih i ponekad konfliktnih informacija. Ono što je ovde zanimljivo napomenuti je da su ti zaključci samo privremeni, odnosno da postoji mogućnost da će se zaključak opovrgnuti, ukoliko se dođe do novih delova informacija.

Znanje se, po *Defeasible* teoriji, deli na činjenice i pravila, kao i relacije superiornosti. Činjenice se u ovom kontekstu posmatraju kao neosporne izjave. Pravila se dalje dele na:

* striktna pravila (eng. *strict* *rules*) – pravila u klasičnom smislu: kad god je premisa neosporna, tada je i zaključak takođe neosporan,
* defeasible pravila – pravila koja se mogu poništiti odgovarajućim dokazima,
* poništavače (eng. *defeaters*) – pravila kojima se ne može doći do bilo kakvog zaključka, koriste se da spreče donošenje zaključaka.

Sama logika se klasifikuje kao “skeptična”, što znači da ne podržava kontradiktorne zaključke, odnosno teži ka tome da razreši konflikte između zaključaka. Samim tim, dolazi se do 4 moguća tipa zaključaka:

* pozitivan definitivni zaključak – zaključak je dokaziv samo uz pomoć činjenica i striktnih pravila (oznaka +D ispred zaključka u SPINdle rezoneru),
* negativan definitivni zaključak – zaključak nije dokaziv samo uz pomoć činjenica i striktnih pravila (oznaka -D ispred zaključka u SPINdle rezoneru),
* pozitivan *defeasible* zaključak – zaključak je defeasible dokaziv (oznaka +d ispred zaključka u SPINdle rezoneru),
* negativan defeasible zaključak – zaključak nije defeasible dokaziv (oznaka -d ispred zaključka u SPINdle rezoneru).

# Predstava zakona u LegalRuleML-u

Različite odredbe u Zakonu o bezbednosti saobraćaja na putevima [7] se mogu predstaviti na različite načine u LegalRuleML jeziku. Pošto su za potrebe ovog rada korišćeni samo članovi koji se odnose na pravo prvenstva prolaza, primećene su izvesne sličnosti u načinu njihovog predstavljanja u LegalRuleML jeziku. Ovo je rezultiralo manjom složenošću parsera, ali zato neki od oblika zakonskih odredbi predviđeni LegalRuleML jezikom nisu podržani.

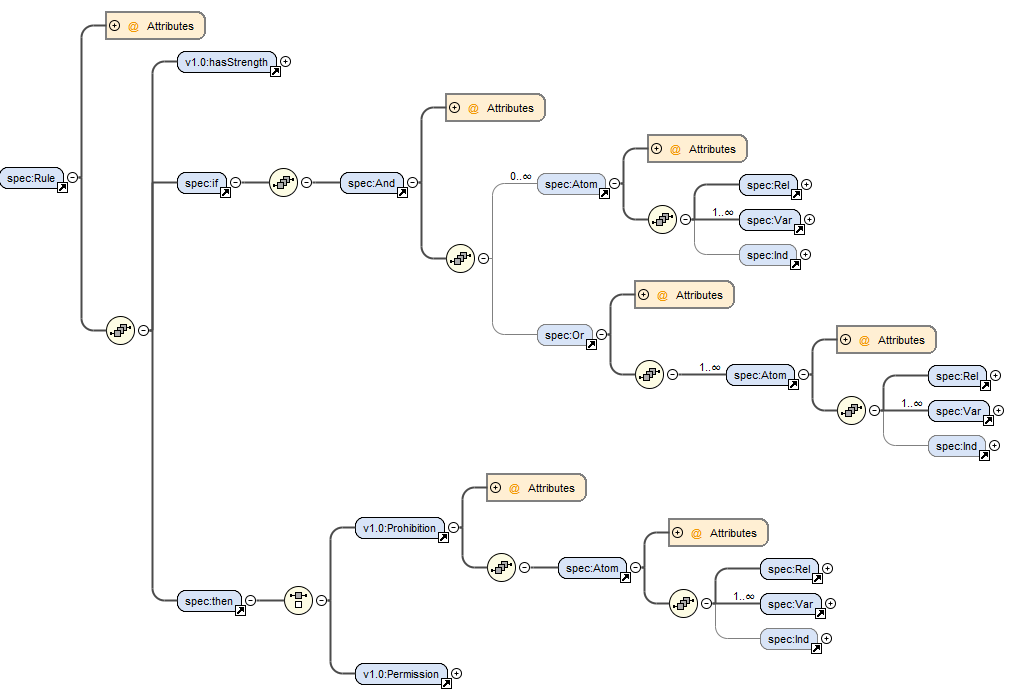
Šablon po kojem su predstavljeni stavovi u LegalRuleML jeziku je sledeći: U okviru <lrml:Statements> elementa postoji mogućnost korišćenja relacije superiornosti sa <lrml:Override> elementom. Zatim se definiše *prescriptive statement* u okviru koga je <lrml:Rule> element, upotrebom <lrml:PrescriptiveStatement> elementa, gde se zatim sam stav predstavlja u vidu uslovnog izraza (ako → onda) gde se koriste elementi <ruleml:If> i <ruleml:Then>. Više opcija je moguće predstaviti u okviru <ruleml:Or> elementa. U okviru <ruleml:Then> elementa je moguće navesti elemente za zabrane (<lrml:Prohibiton), obaveze (<lrml:Obligation>) i dr.

# Metod

Kako bi se došlo do zaključaka prvo je potrebno da se zakonske odredbe predstave u formi LegalRuleML jezika. Zatim se taj LegalRuleML dokument transformiše u RelaxNG šemu [5], koja se potom transformiše u XSD šemu. Iz XSD šeme se dalje generišu Java klase iz kojih se pomoću parsera dobija format *defeasible* logike pogodan za rezonovanje. Za rezonovanje se koristi SPINdle rezoner koji formira zaključke.

Kako je SPINdle rezoner implementiran za rad sa *defeasible* logikom, bilo je potrebno napraviti parser, koji će vršiti prevođenje zakona iz LegalRuleML formata u defeasible logiku.

Ono što ovaj parser razlikuje od većine, je to što ne radi puko parsiranje teksta u dokumentu. Pri izradi se težilo ka očuvanju informacione strukture stabla iz LegalRuleML dokumenta.



Slika – Deo informacione strukture stabla LegalRuleML dokumenta

Da bi se ovo izvelo, bilo je potrebno posmatrati sam LegalRuleML dokument kao XML dokument, što on, na dovoljno visokom nivou apstrakcije, i jeste. Na ovaj način se obezbeđuje uspešno generisanje Java klasa, sa očuvanim svim informacijama iz dokumenta, kao i veza između instanci stabla.

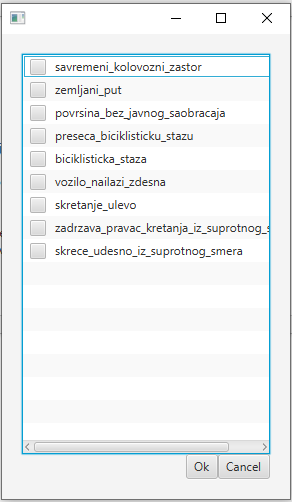
Kako je ovo i dalje specijalizovani RuleML, a samim tim i XML dokument, dobijanje XML šeme iz ručno modelovanog LegalRuleML dokumenta klasičnim metodama se nije pokazalo u potpunosti uspešnim. Kao adekvatnije rešenje pokazalo se korišćenje Relaxngcc [5] *third-party* softvera, koji LegalRuleML dokument prevodi u *Relax NG* šemu, a zatim korišćenje *Trang* softvera [6] za prevođenje tog formata u format XML šeme.

Kada je jednom dobijena potpuna XML šema početnog LegalRuleML dokumenta, prosto korišćenje ugrađenih Java biblioteka dovelo je do generacije neoštećene informacione strukture stabla u formatu Java klasa. Parsiranjem se na osnovu informacija sadržanih u LegalRuleML dokumentu instanciraju objekti generisanih Java klasa, a zapisivanjem tih informacija na adekvatan način dolazi se do pravila reprezentovanih na način pogodan za rezonovanje.   
Taj format *defeasible* logike koji je pogodan za rezonovanje sastoji se od jednog *defeasible* pravila i određenog broja *defeater* pravila. Parser podržava i relaciju superiornosti tako da je moguće da i *defeater-i* budu deo formata *defeasible* logike nad kojima se rezonuje. Deo formata su i činjenice koje je moguće iskoristiti za testiranje različitih slučajeva.

Kako bi se izveli zaključci, koriste se postojeće metode *open source* verzije SPINdle rezonera tako što se nad dobijenim formatom *defeasible* logike pozivaju metode koje omogućuju izvođenje zaključaka. SPINdle rezoner je integrisan u aplikaciju u JAR formatu.

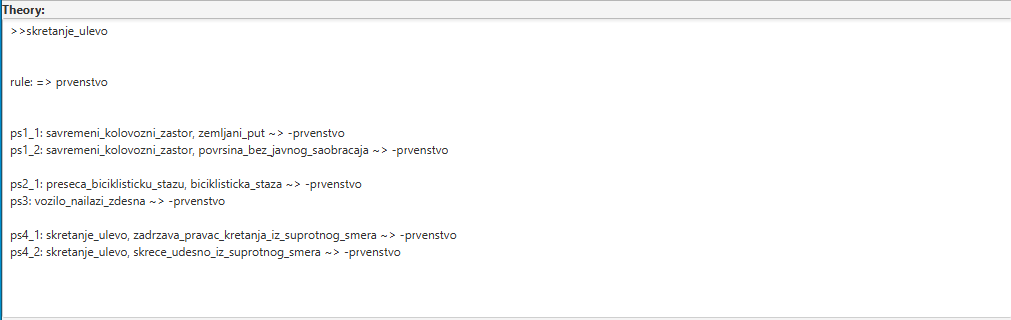
# Rezultati

Sistem koji je razvijen sastoji se od toga da se nad pravilima koji su ručno izvedeni iz zakona u formatu LegalRuleML jezika izvode transformacije pomoću parsera kako bi se dobio format pogodan za rezonovanje. Za rezonovanje se koristi SPINdle rezoner. Ceo sistem je deo aplikacije koja je razvijena za potrebe testiranja. Postupak za korišćenje aplikacije je sledeći. Prvo je potrebno ručno napisati član zakona nad kojim želimo da rezonujemo u formatu LegalRuleML jezika. Zatim se u okviru aplikacije bira taj dokument pritiskom na dugme „Choose LRML file” (Slika 1), koji se automatski propušta kroz parser i dobija se novi dokument u formatu *defeasible* logike. Zatim se bira taj dokument, pritiskom na dugme „Choose DFL file”, što se takođe vidi na slici 1, gde se prvo daje mogućnost odabira činjenica na osnovu kojih će se rezonovati, što se vidi na slici 5.

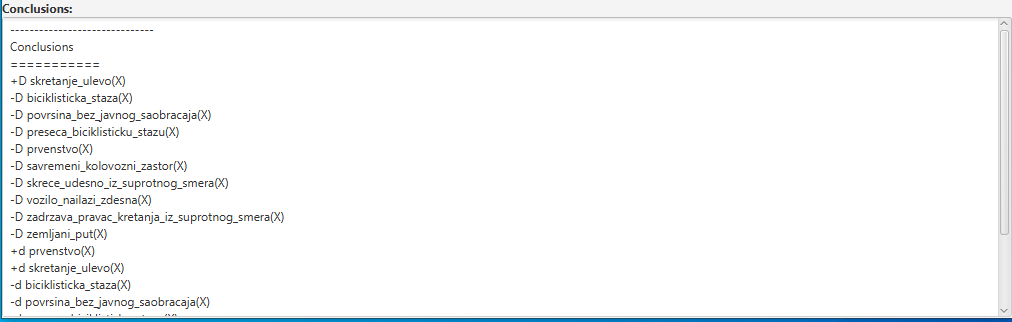


Slika - Odabir činjenica kroz GUI

Pomoću njih se zapravo predstavlja određeni slučaj za koji se primenjuju ta pravila. Na kraju se prikazuje format *defeasible* logike koji je dobijen parsiranjem (Slika 5), kao i izvedeni zaključci (Slika 6). Zaključci su u formatu koji je opisan u poglavlju „Srodna istraživanja“.



Slika - Prikaz defeasible logike



Slika - Prikaz izvedenih zaključaka

# Zaključak

Predstavljen je sistem za zaključivanje baziran na pravilima predstavljenim u LegalRuleML jeziku. Sistem je dizajniran tako da prihvata zakone u LegalRuleML formatu, koji se potom prevodi u *defeasible* logiku, na osnovu koje SPINdle rezoner izvodi zaključke.

Kao glavni doprinos ovog sistema izdvaja se parser LegalRuleML dokumenata, iz razloga što ne radi čisto parsiranje teksta, već teži ka tome da sačuva celokupnu semantiku dokumenta, kao i sve veze između instanci bipartitnog stabla dokumenta. U toku pisanja ovog rada, nije pronađen nijedan parser ovog tipa. Predloženo rešenje može poslužiti i kao komponenta u razvoju softvera autonomnih automobila.

Jedan od izazova za korisnika softvera jeste priprema baze znanja. Od korisnika se očekuje dobro poznavanje LegalRuleML jezika i osnovno poznavanje zakonskih odredbi koje se obrađuju. Potrebno je da ume da prepozna koje se zakonske odredbe mogu napisati u obliku pravila na osnovu kojih se dalje može zaključivati.

Primećeno je da se zakonske odredbe u obliku pravila mogu napisati po određenom šablonu, pa bi tako jedna od olakšica za korisnika mogla da bude pomoćni softver koji bi generisao LegalRuleML dokument u predefinisanom formatu na osnovu zakonskih odredbi koje se obrađuju.

Sistem je moguće proširiti tako da podrži i druge zakone, dok je sada implementiran samo za odabrane članove Zakona o bezbednosti saobraćaja na putevima.

Sledeći korak u ovom pravcu moglo bi biti razvijanje sistema koji bi rezonovao direktno nad LegalRuleML-om. Ovaj pravac bi bio posebno zanimljiv, zato što otvara

mogućnost rezonovanja nad RuleML-om uopšte, čime se otvara polje Semantičkog veba u velikoj meri.

##### Reference

1. OASIS, “LegalRuleML Core Specification Version 1.0”, 2020, dostupno na: http://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/legalruleml-core-spec-v1.0.html (posećeno 14. aprila 2020.)
2. H. Boley, A. Paschke, and O. Shafiq “RuleML 1.0: The Overarching Specification of Web Rules”, International Workshop on Rules and Rule Markup Languages for the Semantic Web, Springer, 2010
3. H. Lam, and M. Hashmi “Enabling Reasoning with LegalRuleML”, Theory and Practice of Logic Programming 2018
4. G. Govematori, “Defeasible Logic”, 2008, dostupno na: <http://www.defeasible.org/> (posećeno 15. aprila 2020.)
5. J. Clark, “RELAX NG Compact Syntax”, 2002, dostupno na: <https://relaxng.org/compact.html> (posećeno 15. aprila 2020.)
6. J. Clark, “Trang, multi-format schema converter based on RELAX NG”, 2008 dostupno na: <https://relaxng.org/jclark/trang.html> (posećeno 15. aprila 2020.)
7. Zakon o bezbednosti saobraćaja na putevima, "Sl. glasnik RS", br. 41/2009, 53/2010, 101/2011, 32/2013 - odluka US, 55/2014, 96/2015 - dr. zakon, 9/2016 - odluka US, 24/2018, 41/2018, 41/2018 - dr. zakon, 87/2018 i 23/2019