

УДК: 340.115:007 ; 004.421.2:512.54.05]:34

CERIF: S 144

ТИП РАДА: ПРЕГЛЕДНИ НАУЧНИ РАД

DOI: 10.55836/PiP_23205A

др *Предраг Н. ЦВЕТКОВИЋ**
редовни професор Универзитета у Нишу – Правног факултета,
Србија

АЛГОРИТМИРАЊЕ ПРАВА: ИЛУСТРАЦИЈА МОГУЋИХ ПРИСТУПА**

Сажетак

Алгоритмирање је процес који омогућава да се тексти нормe преведе у форми који објашњава логичку структуру обавеза и услова одговорности створена на начин који обезбеђује валидност и ефективност уговора који је предмет кодирања. Пројекти развоја платформи и програма за алгоритмирање права засновани су на симбиози програмског кода и правне прозе: у анализираним примерима, Лексон то чини редуковањем правила природног језика на она која могу да буду преведена у код; OpenLaw је заснован на Рикардијанском концепту (део уговора је у правној прози, а део у коду), док Legalese користи језик симболичке логике за превођење сведених правних формулација у форми читљив за програмере. Наведена симбиоза води инжињерисању права и програмског кода на начин који резултира новим форматом правног језика са оригиналном онтологијом. Наведена онтологија заслужује легитимизацију као предмет правног изражавања у циљу њеној будуће свеобухватној, мерљивој и верификабилној ушемељења.

* Електронска адреса аутора: peri@prafak.ni.ac.rs.

** Рад је резултат истраживања на пројекту „Одговорност у правном и друштвеном контексту“, који финансира Правни факултет Универзитета у Нишу.

Кључне речи: *Право као код. – Алгоритмирање права. – Рикардијански уговор. – Онџологија права.*

I Увод

Правне норме написане природним језиком могу бити предмет алгоритмске конверзије у одређеним фазама трајања уговора (спровођење, праћење, контрола, тумачење). Алгоритмирање је процес који омогућава да се текст норме преведе у формат који је разумљив за програмере који развијају програмски код који би требало да омогући извршавање уговора: са тим разумевањем приступили би програмирању правне норме.¹ Алгоритам би требало да објасни логичку структуру обавеза и услова одговорности страна на начин који обезбеђује валидност и ефективност уговора који је предмет кодирања.²

Алгоритмирање уговора је појавни облик преклапања права и програмског кода. Технологија расте експоненцијално, али знање о технологији расте линеарно: стога наведени развој има реметилачки (енг. *disruptive*) ефекат на све области друштва, па и на право. Теза о линеарном расту знања о технологији посебно је квалификована када се ради о правницима који су по дефиницији склони догматизму и мање спремни за искорак. Инфлација правних извора различитог порекла (националних, међународних, супранационалних и слично) генерише све обимније и сложеније правне задатке. Ове задатке могу обављати правници: ипак, примена технолошких решења биће ефикаснија и исплативија на дуге стазе. Одређени правни задаци или процедуре (нпр. тзв. „дужна пажња“ у анализи докумената и процедура – енг. *due diligence*) постављају пред правнике обавезу анализе велике количине података.

1 У том циљу предлаже се коришћење следећих методологија: псеудокода, формалне логике и дијаграма тока (енгл. „*flowchart*“). Вид. више у Предраг Н. Цветковић, „Уговор као алгоритам: уводна разматрања“, *Зборник радова Правног факултета у Нишу*, бр. 92/2021, 15–34. Веза логике и права, те могућност претварања норме у скуп логичких симбола препозната је 50-тих година прошлог века. Вид. Layman E. Allen, „Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents“, *The Yale Law Journal*, Nr. 6/1956, 833.

2 Алгоритмизација се у овом раду анализира кроз визуру претварања уговора у програмски код као *de lege lata* процеса који је видљив и подобан за праћење и истраживање. То не искључује могућност да у блиској будућности буде могуће кодирање прописа и закона. Вид. пример Новог Зеланда у Tom Barraclough, Fraser Hamish and Curtis Barnes, *Legislation as Code for New Zealand*, Auckland, 2021, доступно на адреси: <http://www.nzlii.org/nz/journals/NZLFRRp/2021/3.pdf>, 21. 2. 2023, 175.

Преклапање права и технологије има за резултат настанак *Legal Tech*-а: индустрије која примењује дигиталну технологију на правну област (назив означава технологију примењену на право; енгл. *Legal Technology*). У економском смислу, *LegalTech* је део привреде који повезује технолошко тржиште са тржиштем правних услуга.³

II Алгоритмирање права: типологије приступа

1. Типологија 1: примена телеолошког критеријума

Применом телеолошког критеријума могу се разликовати два приступа развоју претварања/интегрисања права у код.

Први приступ је претварање правне норме у код у циљу вишег нивоа доступности правних услуга; правна норма добија формат кода да би ниво доступности и јасноће правних услуга био виши него што је то случај код класичног, „аналогног“ пружања услуга правног карактера, чиме се у крајњем доприноси „демократизацији“ наведеног сектора. Пример описаног приступа је компанија *Edilex*. *Edilex* развија технологију аутоматизације израде уговора и образаца правних докумената. Циљ *Edilex-a* је поједностављивање правног промета и доступност правних услуга.⁴

Други приступ је рад на програмима унапређења рада правних тимова (адвокатска канцеларија, адвокати, тужилаштва и слично). Пример овог приступа је компанија *Genie AI* која израђује уговоре на основу примене машинског учења.⁵ Користећи машинско учење, софтвер препоручује клаузуле које помажу правницима да ефикасније

3 *LegalTech* заснива се на коришћењу савремених информатичких технологија у области права (правној пракси, пре свега). Вид. Предраг Н. Цветковић, „Примена технологије у правном контексту – пример *LegalTech-a*“, *Право и њивреда*, бр. 3/2022, 447–460.

4 Вид. <https://www.edilex.com/en/>, 21. 3. 2023.

5 Машинско учење (енгл. *Machine Learning*; даље и „МЛ“) је грана вештачке интелигенције чији је циљ конструисање алгоритама и рачунарских система који су способни да се адаптирају на аналогне нове ситуације и уче на бази искуства. МЛ је вештачка интелигенција која се усредсређује на податке, при чему рачунар користи алгоритме уграђене у софтвер да из тих података учи и да се аутоматски побољшава кроз искуство тог учења. Циљ МЛ алгоритама је да се изгради модел заснован на узорку података који може доносити одлуке или предвиђања о новим, раније непознатим узорцима података, а да није експлицитно програмиран да то уради. МЛ се користи у широком спектру апликација попут препознавања слика или говора, аутоматско превођење језика, управљање аутономним превозним средствима (средствима без возача).

састављају уговоре. Програм користи правни језик који је „прикладан за адвокате“⁶

Обе компаније заснивају се на потпуној аутоматизацији: интегришући вештачку интелигенцију у процес састављања уговора, удаљавају се од „статичних“ софтвера попут *Microsoft Word*-а и *Adobe PDF*-а.⁷

2. Типологија 2: процедурални и декларативни програми

Данас постоји велики број програмских језика који су настали у последњих неколико година. Два најпознатија језика на које упућују програмери када предлажу правницима и „ван“ програмерски оријентисаним експертима одакле да започну са учењем програмирања јесу Пајтон (енг. *Python*)⁸ и Пролог.⁹ Они користе различите методе рада. Први је процедурални, други је декларативни.

Процедурални програми одређују како се проблем решава у корацима: постоје јасна упутства која програм треба да прати. Сви појмови су експлицитно дефинисани и сва правила морају бити формулисана. Ако програм, као што је Пајтон, открије да не може да настави са задатком, то је обично зато што програм није у стању да препозна синтаксу Пајтона као програмског језика. Стога је програмирање у Пајтону сензитивно на промене у коду; погрешно постављен зарез или вишак/мањак размака могу утицати на успешност програмирања.¹⁰ Процеду-

6 Вид. <https://genieai.co/home>, 21. 2. 2023.

7 *MS Word* није суштински променио употребу природног језика за креирање уговора. *MS Word* и *Adobe pdf* су технолошки споменици који су дигитална верзија оловке и папира: они немају алгоритамску самосталност, већ служе као нова верзија старог метода записивања правне прозе. Правна заједница почиње да истражује проблеме повезане са употребом статичких платформи као што је *MS Word*. Јуро (<https://info.juro.com>, 23. 2. 2023) је компанија која развија управљање уговорима тако што се *MS Word* контролише преко кода. Олакшавајућа околност овог процеса је што *MS Word* већ има бројне обрасце који су повољни за писање правних докумената: аутоматски унос текста, специјално формирање и слично.

8 <https://www.python.org/>, 21. 3. 2023.

9 ПРОЛОГ је скраћеница од назива „*PROgramming in LOGic*“ То је декларативни програмски језик развијен је са циљем да омогући програмирање путем природног језика. Вид. Марко Марковић, *Инићерактивно састављање машински читљивих и разумљивих судских йисмена базирано на знању*, докторска дисертација, Универзитет у Новом Саду – Факултет техничких наука, Нови Сад, 2018, 33.

10 Рани програмски језици су били процедурални – зато што би програмер дефинисао скуп процедура, које би рачунар потом обрађивао. Процедурално програмирање користи листу инструкција за рачунар које се предузимају корак по корак. Процедурално програмирање се ослања на одређене поступке. Већина раних програмских језика је процедурална. Примери процедуралних језика укључују

рални програми користе самосталне модуле кода којима се може манипулисати и поново користити за друге програмерске задатке.¹¹

Декларативни програми специфицирају проблем и траже од система да га реши. Декларативни језици су засновани на односима (релацијама) између објеката или релацијама између објеката и њихових својстава. Ови односи се могу дефинисати имплицитно кроз правила или експлицитно кроз чињенице: чињенице описују односе, док их правила квалификују. Сврха Пролога да формира фиксни скуп података који би извео одговоре на будућа питања о релацији или скупу релација на основу унетих информација. Насупрот томе, сврха Пајтона је да заврши одређени задатак: сваки корак је експлицитно изражен.

III Примери алгоритмирања права

Пајтон и Пролог су постали инспирација за нову еру програмских језика који се користе за израду аутоматизованих алгоритмираних уговора: *Lexon*, *OpenLaw* и *Legalese*. У наставку следи преглед основних карактеристика наведених програмских језика.

1. Лексон (*Lexon*)

Лексон¹² је програмски језик заснован на лингвистичкој структури и дизајниран да алгоритмира проблем природним језиком. Ово

Fortran, *COBOL* и *C*, који су присутни још од шездесетих и седамдесетих година, а поред њих ту су и *BASIC* и *Pascal*.

11 Теорија модуларности потиче из кибернетике. Временом је постала централни принцип дизајна рачунарских система. Модуларност доприноси да се ефикасно управља сложености комплексних система. Присуство интеракција чини да промена једног елемента (чвора) система изазива каскадне промене у целом систему. Модуларност омогућава да се сложени системи разложе на једноставне компоненте са јасним и мерљивим везама: тиме се спречава да мрежна структура постане неуправљив конгломерат међусобних интеракција. Сложеност постаје управљива прекидањем протока информација унутар система чиме се одређени делови тог система (модули) чине мање зависним од других његових делова: само одређене врсте информација могу се пренети из модула у модул, смањујући значај онога што се дешава у другим модулима. У контексту уговора, модуларни приступ је присутан у пракси израде текста споразума: свођење уговора на модуле може да поједностави трансакције тиме што олакшава управљање елементима тих трансакција. Типичан пример модуларности је клаузула о решавању спорова која је валидна без обзира на валидност уговора. Исто важи и за коришћење дефиниција: оне функционишу као модули јер садрже значење термина који се понављају тако да се измене дефиниције могу извршити без измене других уговорних одредби. Вид. Предраг Н. Цветковић, „Уговор као мрежна структура: корак ка приближавању норме и програмског кода“, *Защити́и́а људских и мањинских љрава у Евројском љравном љросјору* (ур. Небојша Раичевић), Књига XI, Ниш, 2022, 118–120.

12 <http://lexon.tech/>, 21. 3. 2023.

постиже редуковањем речника и граматике природног језика на скупове правила. Лексонов основни речник састоји се од дефинисаних „имена“ који се користе за означавање правних правила и објекта регулисања тих правила. Лексон користи реченице на природном језику са субјектом и предикатом (с тим што не користи чланове, који се сматрају сувишним).

Ево примера уговора састављеног у Лехон-у (на енглеском језику):¹³

LEX Payment.

"Payer" is person.

"Payee" is person.

"Payment" is amount.

Payer pays Payment to Payee.

Лексон нема за циљ да ствара потпуне уговоре, нити да их до краја објашњава. Он служи као мост за разумевање између програмера и правника, тиме што природни језик своди на правила која постоје и у програмским језицима. Његов структурални дизајн користи само функционалност: граматичка правила се користе без контекста (eng. *Context Free Grammar*). Правила граматике делују независно од објеката на које се примењују. Уместо тумачења речи или фразе, Лексон значење ограничава на функцију: у контексту уговора, функционалност је опис скупа права, обавеза и одговорности одређене стране без ослањања на првобитно значење речи.¹⁴ Ова промена има трансформативни утицај на саму природу уговора: настанак и извршење уговора се интегришу и доприносе стварању нове онтологије уговора.¹⁵

13 Преузето са: <http://lexon.tech/>, 12. 1. 2023.

14 Према једном од креатора Лексона, Хенингу Дидриху, за остварење стварне функционалности уговора боље је навести листу права и обавеза уговорних страна без позивања на значење речи. Вид. Henning Diedrich, *Lexon: Digital Contracts*, London, 2020, 106.

15 Уговор у традиционалном („аналогном“) смислу је испуњен када су извршене обавезе њиме предвиђене. Стога је циљ програмирања уговора да аутоматизује извршење предвиђених обавеза: циљ није анализа текста или интерпретација уговорних норми. Инструкције програмерима морају да буду прецизне и недвосмислене. Аутоматизује се извршење одређених задатака садржаних у уго-

2. *OpenLaw*

OpenLaw користи језик за означавање текста (енг. *Mark-up language*)¹⁶ у циљу претварања уговора на природном језику у објекте читљиве за програмске језике кроз дефинисање релевантних варијабила и логике функционисања тих варијабила у односном правном тексту. Циљ *OpenLaw* програма није да се уговори о природном језику преведу у целини. Уместо тога, резултат примене је интегрисање програмског кода са уговорним одредбама у форми правне прозе. Ради се о језику који примењује формат Рикардијанског уговора.¹⁷ Намера је да се генеришу

ворним обавезама (нпр. плаћање, испорука, слање обавештења о пријему итд.). Програмирање уговора има за резултат аутоматизацију извршења: ова аутоматизација праћена је гаранцијом да ће извршење у питању бити учињено управо како је уговором усаглашено. За разлику од традиционалног („аналогно“) уговора, који разликује текст уговора и његово извршење, програмски код допушта да текст контракта у програмском језику буде истовремено и акција његовог испуњења. Тиме се потенцијално изједначава уговорно регулисање и испуњење уговора: самим постанком кода (нормирањем) оно што би требало да буде учињено као резултат уговора већ је извршено с обзиром на непроменљивост програмског кода и његову затвореност за интервенцију посредника. Уговорни документ у форми кода више није отелотворење споразума већ средство његовог испуњења. Вид. Lawrence Edward Diver, *Digisprudence: the affordance of legitimacy in code-as-law*, doctoral thesis, University of Edinburgh, Edinburgh, 2019, доступно на адреси: <https://era.ed.ac.uk/bitstream/handle/1842/36567/Diver2019.pdf?sequence=1>, 21. 2. 2023, 304.

- 16 Вид. <https://docs.openlaw.io/markup-language/#variables>, 21. 3. 2023. Техника означавања текста (енг. *Mark-up*) сеже још из периода пре постојања рачунара, када су људи пре штампања текста, ручно, на рукопису означавали на који начин би поједини делови требало да буду одштампани (нпр. којом врстом и величином слова). Слична техника се користи и данас, у ери рачунара, и текст се додатно означава информацијама које га описују. Језици за означавање често, осим што описују визуелну презентацију, описују и логичку структуру документа. Ово је значајно јер се тиме омогућава униформна измена одређених елемената документа (на пример свих наслова), односно додељивање једног својства свим елементима тог типа. Најпознатији језик за означавање је *HTML*, намењен опису веб страница.
- 17 Рикардијански уговор је документ који је: истовремено читљив за људе (као и уговор на папиру) и за програм. Први пут га је представио Јан Григ (*Ian Grigg*) 1996. године. Григ је био криптограф који се сматра пиониром финансијске криптографије. Концепт Рикардијанског уговора настао је током покушаја дигитализације уговора којима се уређују финансијски инструменти. Рикардијански уговор (именован је по Давиду Рикарду у част његовог доприноса теорији међународне трговине) не аутоматизује претпостављене елементе споразума кроз примену програмског кода. Уместо тога, његов је циљ да обезбеди флексибилност текстуалних споразума уз кодирање тамо где је то могуће. Резултат овог процеса да програмски код пре комплентира, него што замењује споразуме у текстуалној форми (примена принципа „више права мање софтвера“). Текст уговора који није у програмском коду требало би да буде формулисан на начин који одговара минималистичкој семантици кода. Описана минималистичка семантика омогућава да програмски код обезбеди гаранцију интегритета информације садржане у коду (информације

варијабле и логика која ће бити уграђена у уговоре одређеног типа. На пример, уговор о чувању пословне тајне (енг. „*Non-Disclosure Agreement*“) обично би узимао имена уговорних страна и трансформисао их као динамичке варијабле. Ако променљива захтева даљи опис, додатни текст (енг. *string*)¹⁸ се користи за квалификацију термина. Карактеристика *OpenLaw* језика је Булова логика:¹⁹ услови (функције), прожимају логику уговора и реконструишу уговорне услове у бинарна питања. Клаузуле се тумаче као „уграђени образац“: циљ је да се формулисање уговора учини ефикаснијим тиме што ће се клаузуле за стандардне верзије чувати као подаци који се могу додати уговорима.

У наставку је дат пример уговорне одредбе из споразума о пружању услуга саветовања формулисан у *OpenLaw* формату.²⁰

```
\centered **Simple Advisor Agreement**

This Advisor Agreement is entered into between [[Company Name]] ("Corporation")
and [[Advisor Name]] ("Advisor") as of [[Effective Date: Date]] ("Effective Date")
Company and Advisor agree as follows:

^**Services.** Advisor agrees to consult with and advise Company from time to time
at Company's request (the "Services").

^[[Choice of Law Insert: Clause("Choice of Law and Venue Clause")]]

^**Termination.** Either party may terminate this Agreement at any time, for any
reason, by giving the other notice.
```

садржане у самом уговору и претворене у програмски језик) и проверу њеног порекла. Тиме се непроменљивост програмског кода комбинује са флексибилношћу изражавања: флексибилност као могућност избора је одређена у тексту, док са друге стране програмски код обезбеђује непроменљивост и интегритет информација које су садржане у писаном споразуму, али су конвертоване у програмски језик (онда када је таква конверзија могућа; одредбе споразума које нису подобне за наведену конверзију садржане су у тексту уговора који представља саставни део Рикардијанског споразума). Вид. Lopamudra Mandal, *Ricardian Contracts – Bringing the Gap Between Smart Contracts and Traditional Contracts*, master thesis, Tilburg University, Tilburg, 2019, доступно на адреси: <http://arno.uvt.nl/show.cgi?fid=149417>, 18. 4. 2023.

- 18 У програмирању *string* је секвенца симбола-карактера и представља текст.
- 19 Булова алгебра је део математичке логике: ради се о структури која се заснива на логичким операцијама „и“, „или“ и „не“. За разлику од елементарне алгебре, где се вредност променљивих означава бројевима, у Буловој алгебри вредности променљивих могу бити само тачно и нетачно (истина и не-истина), што се обично означава са 1 и 0, где 1 представља тачно, а 0 нетачно.
- 20 Вид. <https://docs.openlaw.io/markup-language/#variables>, 12. 3. 2023.

Simple Advisor Agreement

This Advisor Agreement is entered into between [[Company Name]] ("Corporation") and [[Advisor Name]] ("Advisor") as of [[Effective Date]] ("Effective Date"). Company and Advisor agree as follows:

1. **Services.** Advisor agrees to consult with and advise Company from time to time, at Company's request (the "Services").
2. **Choice of Law and Venue.** The parties agree that this Agreement is to be governed by and construed under the law of the State of [[State of Governing Law]] without regard to its conflicts of law provisions. The parties further agree that all disputes shall be resolved exclusively in state or federal court in [[County of Venue]], [[State of Venue]].
3. **Termination.** Either party may terminate this Agreement at any time, for any reason, by giving the other notice.

Горе наведени извод из уговора је представљен у два облика: (1) у коду и; и (2) у формату правне прозе. Претпоставка овог приступа је идентификовање делова споразума који се могу конвертовати у програмски код.

3. *Legalese*

Legalese (даље и „Л4“) је језик специфичан за домен права (енг. „*Domain Specific Language*“ – *DSL*) настао с циљем да уважи карактеристике права његову семантику, деонтику и логику.²¹

Заснован је на концепту/логици Пролога, али је настао са намером да програмира правне норме. Сврха Л4 се протеже даље од опште примене програмских језика на правни језик.

Л4 производи формално верификоване кодирани уговоре који се подједнако могу трансформисати у ПДФ-ове написане на природном језику. Идеја је да је програм обезбеди непрекоран превод уговорних услова са правне прозе на код. Идеја о Л4 потекла је од програмера који су желели да конвертују инвестициони уговор написан на природном језику на језик симбола.²² У наставку је текст дела овог уговора на енглеском језику.²³

21 Вид. <https://legalese.com/aboutus.html#innovation-premise>, 18. 3. 2023.

22 Вид. <https://legalese.com/computational-law.html>, 18. 3. 2023.

23 Преузето са: „Why Computational Law?“, *Legalese*, доступно на адреси: <https://legalese.com/computational-law.html>, 20. 4. 2023.

1.2.1 If the investment for the purpose of the Series B Funding is valued at not more than \$32.5 Million, then the investors in the Note shall be entitled to convert the Note into Shares at a fixed valuation of \$27.5 million.

1.2.2 If the investment for the purpose of the Series B Funding is valued at less than \$40 million but not below \$32.5 million, investors in the Convertible Note will be entitled to convert the Note into Shares at a 15% discount over the valuation of the Series B Funding (for instance, if the series B Funding is at a valuation of \$35 million, then the investors in the Note shall be entitled to convert at a valuation of 35M less 15% discount);

1.2.3 If the investment for the purpose of the Series B Funding is valued at not less than \$40 million but less than \$47.06 million, investors in the Convertible Note will be entitled to convert the Note into Shares at a 15% discount over the valuation of the Series B Funding (for instance, if the series B Funding is at a valuation of \$47.06 million, then the investors in the Note shall be entitled to convert at a pre-money valuation of 40M i.e. \$47.06 million less 15% discount);

1.2.4 If the investment for the purpose of the Series B Funding is valued at not less than \$47.06 million but less than \$80 million, investors in the Convertible Note will be entitled to convert the Note into Shares at a fixed pre-money valuation of \$40 million;

1.2.5 If the investment for the purpose of the Series B Funding is valued at not less than \$80M but less than \$100M, investors in the Convertible Note will be entitled to convert the Note into Shares at a fixed pre-money valuation of \$45M; and

1.2.6 If the investment for the purpose of the Series B Funding is valued at not less than \$100 million, investors in the Convertible Note will be entitled to convert the Note into Shares at a fixed pre-money valuation of \$50 million.

Превод горе наведеног текста коришћењем *Legalese* програмског језика гласи:

```
if( seriesB < 32.5 ) { conversion = 27.5 }
else if( seriesB < 40 ) { conversion = seriesB * 0.85 }
else if( seriesB < 47.06 ) { conversion = seriesB * 0.85 }
else if( seriesB < 80 ) { conversion = 40 }
else if( seriesB < 100 ) { conversion = 45 }
else { conversion = 50 }
```

Понављање именичних фраза у делу уговора у формату правне прозе доводи до тога да за интерпретацију није потребан контекст: довољна је синтакса и функције. Резултат тога је могућност да се природан језик трансформише у јасну алгоритамску/рачунарску форму.

IV Закључак

Правне норме су могући предмет програмирања различитим фазама (дефинисање, спровођење, праћење, контрола, тумачење). Програмирање правних правила је процес претварања правне норме у неки од програмских језика. Алгоритмирање уговора је појавни облик преклапања права и програмског кода. Циљ алгоритмирања је да се обезбеди да текстуалне формулације буду преведене у форму разумљиву програмерима, с коначним ефектом идентитета правног значења норме у правној прози и у форми програмског кода. Горе наведени програмски језици помажу у аутоматизацији процеса креирања и тумачења уговора. Интерпретација је интернализована у техничким границама програмског језика: постоје термини и одредбе које не могу да буду, *de lege lata*, предмет претварања у код с обзиром на ограничења технологије (ради се пре свега о правним стандардима). Сва три примера алгоритмирања правне норме (Лексон, *OpenLaw*, *Legalese*) користе и даље правну прозу као важан сегмент конверзије права у код. Лексон то чини редуковањем правила природног језика на она која могу да буду преведена у код (нпр. изостају чланови уз именице); *OpenLaw* користи Рикардијански концепт (део уговора је у правној прози, а део у коду), док *Legalese* користи језик симболичке логике као платформу за превођење сведених правних формулација у формат читљив за програмере. Интегрисање правне прозе и програмског кода (а не искључиво конверзија текста у код) резултира новим форматом правног језика са оригиналном онтологијом која је резултат приступа „кодирати шта је могуће, написати шта се мора“. Да ли програмски језици треба да буду признати као облик правног језика? Чини се да постоји простор да се овакав приступ легитимно заступа, с тим да предстоји рад на његовом потпуном, јасном и верификованом утемељењу.

Коришћена литература

- Allen Layman E., „Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents“, *The Yale Law Journal*, Nr. 6/1956, 833.
- Barracrough Tom, Hamish Fraser, Bames Curtis, *Legislation as Code for New Zealand*, Auckland, 2021, доступно на адреси: <http://www.nzlii.org/nz/journals/NZLFRRp/2021/3.pdf>, 21. 2. 2023.
- Diver Lawrence Edward, *Digisprudence: the affordance of legitimacy in code-as-law*, doctoral thesis, University of Edinburgh, Edinburgh, 2019, доступно на адреси: <https://era.ed.ac.uk/bitstream/handle/1842/36567/Diver2019.pdf?sequence=>, 21. 2. 2021.

- Diedrich Henning, *Lexon: Digital Contracts*, London, 2020.
- Mandal Lopamudra, *Ricardian Contracts – Bringing the Gap Between Smart Contracts and Traditional Contracts*, master thesis, Tilburg University, Tilburg, 2019, доступно на адреси: <http://arno.uvt.nl/show.cgi?fid=149417>, 18. 4. 2023.
- Марковић Марко, *Интерактивно састављање машински читљивих и разумљивих судских писмена базирано на знању*, докторска дисертација, Универзитет у Новом Саду – Факултет техничких наука, Нови Сад, 2018. (Marković Marko, *Interaktivno sastavljanje mašinski čitljivih i razumljivih sudskih pismena bazirano na znanju*, doktorska disertacija, Univerzitet u Novom Sadu – Fakultet tehničkih nauka, Novi Sad, 2018)
- Цветковић Предраг Н., „Примена технологије у правном контексту – пример *LegalTech*-а“, *Право и привреда*, бр. 3/2022. (Cvetković Predrag N., „Primena tehnologije u pravnom kontekstu – primer *LegalTech*-а“, *Pravo i privreda*, br. 3/2022)
- Цветковић Предраг Н., „Уговор као алгоритам: уводна разматрања“, *Зборник радова Правног факултета у Нишу*, бр. 92/2021. (Cvetković Predrag N., „Ugovor kao algoritam: uvodna razmatranja“, *Zbornik radova Pravnog fakulteta u Nišu*, br. 92/2021)
- Цветковић Предраг Н., „Уговор као мрежна структура: корак ка приближавању норме и програмског кода“, *Заштита људских и мањинских права у Европском правном простору* (ур. Небојша Раичевић), Књига XI, Ниш, 2022. (Cvetković Predrag N., „Ugovor kao mrežna struktura: korak ka približavanju norme i programskog koda“, *Zaštita ljudskih i manjinskih prava u Evropskom pravnom prostoru* (ur. Nebojša Raičević), Knjiga XI, Niš, 2022)

Predrag N. CVETKOVIĆ, PhD

Full Professor at the University of Niš Faculty of Law, Serbia

ALGORITHMIZATION OF LAW: AN ILLUSTRATION OF POSSIBLE APPROACHES

Summary

Algorithmization is a process that enables the text of the norm to be translated into a format explaining the logical structure of the duties and responsibilities of the parties and ensuring the validity and effectiveness of the contract that is the subject of coding. Platforms and programs for algorithmization of law are based on the symbiosis of programming code and legal prose: in the analyzed approaches, Lexon achieves this symbiosis by reducing the rules of natural language to those that can be translated into code; OpenLaw makes it by using concept of Ricardian contract (part of the contract is in legal prose and part in code), while Legalese uses the language of symbolic logic to translate condensed legal wording into a programmer-readable format. The aforementioned symbiosis leads to the integration of law and programming code which results in a new legal language format with an original ontology. The foregoing ontology deserves legitimization as a subject of legal research in order to provide its comprehensive, measurable and verifiable foundation.

Key words: *Law as Code. – Algorithmization of Law. – Ricardian Contract.
– Ontology of law.*

Датум пријема рада: 18. 4. 2023.

Датум прихватања рада: 15. 5. 2023.