

Teisinių žinių bazių situacinio taikomumo analizės metodas

Laima PALIULIONIENĖ (MII)

el. paštas: laipal@ktl.mii.lt

Reziumė. Daugelį žinių bazių verifikavimo ir vertinimo metodų galima naudingai pritaikyti ir teisinių žinių bazėms, tačiau teisės dalykinės srities ypatumai reikalauja ir specifinių metodų. Vienas iš teisės aktų kokybės elementų yra jo vidinė ir išorinė darna. Viena iš darnos rūšių yra kiekybiškai išreikštų reikšmių darna, o taip pat kitokių reikšmių, kurioms galima nustatyti sutvarkymo santykį ir palyginti tarpusavyje, darna. Kiekybiškai išreikštos reikšmės yra, pavyzdžiui, sankcijos už baudžiamosios ir administracinės teisės pažeidimus (baudos dydis, įkalinimo trukmė ir pan.), mokesčiai, pašalpos ir pensijos. Šiame straipsnyje siūlomas metodas, kuris įgalina analizuoti teisės akto taikymo įvairioms galimoms situacijoms pasekmes ir aptikti nedarną tarp kiekybiškai išreikštų rezultatų.

Raktiniai žodžiai: teisinės žinių bazės, legimitika, verifikavimas ir vertinimas, teisinių žinių bazės darna.

1. Įvadas

Dirbtinio intelekto ir kitų kompiuterizuotų metodų naudojimas rengiant teisės aktus gali padėti gerinti jų kokybę. Mokslo sritis, tyrinėjanti kompiuterizuotą įstatymų rengimą, vadinama *legimitika* [1]. Siekiant panaudoti dirbtinio intelekto metodus, teisės aktus reikia formalizuoti ir saugoti žinių bazėse. Žinių bazių kokybę tikrinama verifikavimo ir vertinimo būdu (angl. V&V – verification and validation) [2, 3, 4]. Nors daugelį žinių bazių verifikavimo ir vertinimo metodų galima naudingai pritaikyti ir teisinių žinių bazėms, teisės dalykinės srities ypatumai reikalauja ir specifinių metodų. Pavyzdžiui, [5] aprašytas metodas, skirtas patikrinti įstatymo taikymo adekvatumą, modeliuojant atskirų straipsnių taikymo ir netaikymo situacijas ir pateikiant rezultatus ekspertui analizuoti. Jis padeda išryškinti situacijas, kai straipsnis formaliai gali būti pritaikytas, tačiau toks taikymas nėra pageidaujamas ir nebuvo sąmoningai numatytas įstatymo rengėju.

Vienas iš teisės aktų kokybės elementų yra jo darna – tiek vidinė (vieno teisės akto ribose), tiek išorinė (kitų teisės aktų atžvilgiu). Darna reiškia prieštaravimų nebuvimą ir teisės akto atitikimą nustatytoms ribojimo taisyklėms. Viena iš darnos rūšių yra kiekybiškai išreikštų reikšmių darna, o taip pat kitokių reikšmių, kurioms galima nustatyti sutvarkymo santykį ir palyginti tarpusavyje, darna. Kiekybiškai išreikštos reikšmės yra, pavyzdžiui, sankcijos už baudžiamosios ir administracinės teisės pažeidimus, mokesčiai, pašalpos ir pensijos.

Pavyzdžiui, baudžiamuosiuose įstatymuose kiekybiškai (baudos dydžiu, įkalinimo trukme ir pan.) tenka išreikšti teisingumą kaip vertybinę kategoriją. Sankcijų sistema yra darni, jei numatytos bausmės yra proporcingos nusikalstamų veikų pavojingumui.

Ši sistema dažniausiai yra didelė ir sudėtinga, todėl žmogui sunku aprėpti visus sistemos elementus ir juos suderinti. Vienintelė išeitis – formalizuoti sankcijų nustatymo kriterijus sukūrus specialų nusikalstamos veikos modelį [6] ir panaudoti kompiuterizuotus rengiamų teisės aktų testavimo metodus. Formalizuojant sankcijų nustatymo kriterijus reikia nustatyti vertybių santykinę svarbą. Pavyzdžiui, su fiziniiais asmenimis siejamos vertybės yra gyvybė, sveikata, nuosavybė, garbė ir orumas ir kt. Vertybių svarba nustatoma remiantis tradicijomis, teismų praktika, įvairių šalių teisės aktų lyginamąja analize, ekspertų vertinimais. Tik po to, kai sukurta vertybių sistema ir nusikalstamos veikos modelis, galima kalbėti apie automatizuotą sankcijų sistemos darnos tikrinimą.

Kita vertus, mokesčių, pašalpų ir pensijų skyrimo kriterijai yra aiškesni, nes nereikalauja daugelio vertybių klasifikavimo pagal jų svarbą. Todėl ir darnos tikrinimas šiuo atveju yra paprastesnis.

Šiame straipsnyje siūlomas metodas, kuris įgalina analizuoti teisės akto taikymo įvairioms galimoms situacijoms pasekmes ir aptikti nedarną. Metodas veikia teisės akto struktūrinio vieneto (straipsnio) lygmeniu, testai generuojami atitinkamos ontologijos ir teisės akto koncepcinio modelio pagrindu.

Straipsnyje nagrinėjamas teisinių žinių vaizdavimas, aprašomas siūlomas situacinio taikomumo analizės metodas, pateikiamas metodo taikymo pavyzdys ir išvados.

2. Teisinių žinių vaizdavimas

Mūsų tyrime žinioms vaizduoti naudojamas freimų logikos formalizmas F-logika [7]. Šis formalizmas jungia objektinį žinių vaizdavimo būdą su loginio išvedimo mechanizmu, grindžiamu rezoliucija. F-programą sudaro taisyklės, kurių forma yra *antraštė* \leftarrow *kūnas*, kur *antraštė* (taisyklės konsekventas) yra F-molekulė, *kūnas* (taisyklės antecedentas) yra F-molekulių ir jų neiginių konjunkcija. Pagal F-logikos sintaksę, konstantos prasideda mažąja raide, kintamieji – didžiąja raide. F-molekulėse naudojami klasių ir jų egzempliorių vardai, atributų vardai, atributų reikšmės, predikatų simboliai. Klasių, egzempliorių ir atributų vardai yra identifikaciniai termai, kurie gali turėti argumentus. Žemiau pateiktas klasės „asmuo“ formalizavimo pavyzdys, kuris bus naudojamas toliau nagrinėjant metodo taikymo pavyzdį, ir klasės „invalidumo_grupė“ egzempliorių sukūrimo pavyzdys („grupė(1)“, „grupė(2)“ ir „grupė(3)“ yra identifikaciniai termai su argumentais, kurie reiškia pirmą, antrą ir trečią invalidumo grupę).

```
asmuo [pavardė => string; sodra_nr => string; vaikų_skaičius => integer;
      invalidumas => invalidumo_grupė; ar_vienišas => boolean;
      ar_žemės_ūkio_darbuotojas => boolean].
grupė(1): invalidumo_grupė.
grupė(2): invalidumo_grupė.
grupė(3): invalidumo_grupė.
```

F-molekulės sudedamosios dalys turi vieną iš šių pavidalų: (a) $S_1 : S$, kur S_1 žymi klasės S egzempliorių; (b) $S_1 :: S$, kur S_1 žymi klasės S poklasį; (c) $S[K \rightarrow R]$, kur S žymi klasės egzempliorių, K – atributą, R – to atributo reikšmę; (d) $S[K \Rightarrow R]$, kur S žymi klasę, K – atributą, R – to atributo reikšmių kitimo sritį; (e) $S[K * \rightarrow R]$, kur

S klasę, K – atributą, kurio reikšmę galima naudoti klasės egzemplioriuose neišreikštinu būdu, R – to atributo reikšmę, kuri pagal nutylėjimą perduodama klasės egzemplioriams; (f) $P(S_1, \dots, S_n)$, kur P yra predikato vardas, S_1, \dots, S_n – identifikaciniai termai, žymintys klases, egzempliorius, atributus, atributų reikšmes. Atributai gali būti daugiareikšmiai; tuomet naudojami ženklai $->>$, $=>>$ ir $*->>$. Kadangi atributų reikšmės F-logikoje gali būti perduodamos pagal nutylėjimą, joje galima modeliuoti nemonotonines žinias. F-logikos dalis be tokio atributų reikšmių perdavimo yra ekvivalentiška predikatų skaičiavimui, t. y. tarp jų egzistuoja abipusis atvaizdavimas.

Formalizuojant žinias, tikslinga naudoti ontologijas [8]. Ontologijos aprašo bendrines arba dalykinės srities sąvokas, šių sąvokų sąryšius (visų pirma taksonominius, bet ne tik) ir aksiomas, kurios riboja apibrėžtų sąvokų, sąryšių interpretavimą [9]. Sąvokos ir jų prasmė įvairiuose teisės aktuose gali skirtis. Ontologijos naudojamos kuriant konkrečių įstatymų koncepcinius modelius. Sąvokas galima aprašyti naudojant F-logikos vaizdavimo formalizmą.

Mūsų koncepciniame modelyje kiekvienas teisės akto straipsnis aprašomas kaip klasės „straipsnis“ egzempliorius, o straipsnio tekstas formalizuojamas taisyklėmis. Kiekvieno straipsnio taisyklės įrašomos atskirame modulyje:

```
module M{
  M: straipsnis [nr ->NM; pavadinimas ->PM].
  <taisyklės>
}
```

Čia M yra straipsnio vardas koncepciniame modelyje, N_M – straipsnio numeris, P_M – straipsnio pavadinimas.

Nagrinėsime taisyklės, kuriose sąlygos ir išvados yra susijusios su vienos klasės atributais. Pavyzdžiui, klasės „asmuo“ atributai gali aprašyti tiek pradines savybes (pvz., amžių, invalidumą, vaikų skaičių ir pan.), tiek išvestinius rezultatus (pvz., skiriamą pensiją, taikomą neapmokestinamą pajamų dydį ir pan.), pavyzdžiui:

```
X[inpd -> 430] <-X: asmuo [invalidumas -> grupė(1)].
```

Čia atributas „inpd“ reiškia individualų neapmokestinamą pajamų dydį.

3. Teisinių žinių bazių situacinės analizės metodas

Siūlomas dviejų lygmenų metodas, kuris įgalina analizuoti teisės akto taikymo įvairioms galimoms situacijoms pasekmes ir aptikti nedarną tarp kiekybiškai išreikštų rezultatų. Pirmajame lygmenyje automatizuotu būdu nustatomi teisės akto taikymo įvairiose galimose situacijose rezultatai. Šie rezultatai lentelės pavidalu pateikiami eksperto vertinimui. Kartais nedarnai aptikti pakanka tokio vizualinio vertinimo, tačiau jei galimų situacijų yra daug, žmogus gali ir nepastebėti klaidos. Tuomet reikalingas antrasis metodo lygmuo, kuriame naudojamas formaliai užrašytas darnos ribojimas, kuris tikrinamas automatiškai.

Metodas skirtas taikyti straipsniams, kurių sąlygos ir išvados yra susijusios su vienos klasės atributais, kaip aprašyta aukščiau. Žemiau pateiktos abiejų metodo lygmenų užklauso formos ir algoritmai. Šios užklauso nėra F-logikos užklauso, todėl reikalingas papildomas užklauso kalbos transliatorius.

1. Pirmasis lygmuo

Užklauso forma:

?-test module(M) class(K) methods(A_1, A_2, \dots, A_n) result-method(R).

Tai užklauso apie straipsnio taikymo rezultatus: „Kokios bus klasės K atributo R reikšmės priklausomai nuo atributų A_1, A_2, \dots, A_n reikšmių atliekant išvedimą teisės akto struktūriniame vienete M ?“

Kitas šios užklauso variantas leidžia tiesiogiai nurodyti kai kurių atributų testuojamas reikšmes:

?-test module(M) class(K) methods($A_1(V_{11}, V_{12}, \dots, V_{1m_1}), \dots,$
 $A_n(V_{n1}, V_{n2}, \dots, V_{nm_n})$) result-method(R).

Algoritmas:

Pirmasis žingsnis. Sugeneruoti testus (testuojamąsias situacijas) sukuriant klasės K egzempliorius su įvairiais atributų A_1, A_2, \dots, A_n reikšmių deriniais.

K -egz: $K[A_1 \rightarrow V_1; A_2 \rightarrow V_2; \dots; A_n \rightarrow V_n]$.

Čia V_i priklauso atributo A_i reikšmių kitimo aibei. Testams skirtas skaitines reikšmes (pavyzdžiui, „integer“ arba „real“ tipo atveju) turi išreikštinu būdu nurodyti žinių inžinierius. Neskaitiniai atributai teisės akto koncepciniame modelyje apibrėžiami baigtine galimų reikšmių aibe, o ne intervalais, todėl jų reikšmes galima generuoti automatiškai iš koncepcinio modelio.

Antrasis žingsnis. Papildyti žinių bazę vienu iš testų ir atlikti išvedimą atsakant į užklauso:

?- $K[R \rightarrow X]$.

Čia K ir R yra metakintamieji, kurie konkrečioje užklausoje keičiami į konkrečius klasės ir atributo vardus, o X yra užklauso kintamasis, kurio reikšmę reikia išvesti.

Šis žingsnis kartojamas su visais pirmajame žingsnyje sugeneruotais testais.

Trečiasis žingsnis. Pavaizduoti rezultatus lentelės pavidalu taip, kad ekspertui būtų patogų juos analizuoti.

2. Antrasis lygmuo.

Užklauso forma

Tokia pat, kaip pirmajame lygmenyje, tik prisideda dėmuo constraint (C), kuris nurodo, kuri ribojimą reikia tikrinti:

?-test module(M) class(K) methods(A_1, A_2, \dots, A_n) result-method(R) constraint(C).

Pats ribojimas žinių bazėje užrašomas taip:

ribojimas [tipas => string]

C : ribojimas [tipas ->> C_tipas] <- <ribojimo pažeidimo sąlygos>

Pirmasis žingsnis. Toks pat, kaip pirmajame lygmenyje.

Antrasis žingsnis. Toks pat, kaip pirmajame lygmenyje.

Trečiasis žingsnis. Sukurti laikiną žinių bazę, kurią sudarytų C ribojimo taisyklė ir klasės K egzemplioriai, sukurti pirmame žingsnyje, kuriuose būtų ir atitinkama atributo R reikšmė, išvesta antrame žingsnyje, t.y. K -egz: $K[A_1 \rightarrow V_1; A_2 \rightarrow V_2; \dots; A_n \rightarrow V_n; R \rightarrow X]$.

Ketvirtasis žingsnis. Laikinojoje žinių bazėje atlikti išvedimą atsakant į užklausą ?–C[tipas →Y]. Teigiamas išvedimo rezultatas reiškia, kad aptiktas ribojimo pažeidimas. Tuomet ekspertui pateikiamos ir ribojimo pažeidimo sąlygos reiškinys su konkretizuotais kintamaisiais, sukėlusiais ribojimo pažeidimą.

4. Metodo taikymo pavyzdys

Panagrinėkime pavyzdį, susijusį su tokiu Gyventojų pajamų mokesčio įstatymo straipsniu:

20 straipsnis. Neapmokestinamasis pajamų dydis ir papildomas neapmokestinamasis pajamų dydis

1. Pagrindinis neapmokestinamasis pajamų dydis (pagrindinis NPD) – 290 litų per mėnesį.
2. Šiems nuolatiniais Lietuvos gyventojams taikomi individualūs neapmokestinamieji pajamų dydžiai (jeigu gyventojas atitinka ne vieną iš 1–5 punktuose nustatytų kriterijų, taikomas didžiausias individualus NPD):
 - 1) I grupės invalidams – 430 litų per mėnesį;
 - 2) II grupės invalidams – 380 litų per mėnesį;
 - 3) asmenims, auginantiems tris ir daugiau vaikų (įvaikių) iki 18 metų [...] – 430 litų per mėnesį, be to, už ketvirtą ir kiekvieną paskesnę vaiką (įvaikį) NPD didinamas 46 litais;
 - 4) motinai (motei) arba tėvui (tėviui), kuris (kuri) vaikus (įvaikius) iki 18 metų [...] auginą vienas (viena), – 335 litai per mėnesį, be to, už auginamą antrą ir kiekvieną paskesnę vaiką (įvaikį) NPD didinamas 53 litais;
 - 5) žemės ūkio veiklos subjektų darbuotojams [...] – 330 litų per mėnesį.
6. Nuolatiniais Lietuvos gyventojams (tėvams arba tėviams), auginantiems vieną ar du vaikus (įvaikius) iki 18 metų [...] už kiekvieną auginamą vaiką (įvaikį) yra taikomas papildomas neapmokestinamasis pajamų dydis (toliau – PNP), kuris lygus 0,1 pagrindinio NPD. Šios dalies nuostatos netaikomos šio straipsnio 2 dalies 4 punkte nurodytiems nuolatiniais Lietuvos gyventojams.
7. Mokestiniai laikotarpiai PNP taikomas toje pajamų, susijusių su darbo santykiais arba jų esmė atitinkančiais santykiais, gavimo vietoje, kur taikomas NPD, dalijant PNP sumą kiekvienam iš tėvų (tėvių) per pusę...

Šis įstatymo straipsnis žinių bazėje formalizuojamas taip:

```

module s20 {
s20: straipsnis [nr → 20; pavadinimas → „Neapmokestinamasis pajamų dydis ir papildomas
neapmokestinamasis pajamų dydis“].
X[inpd *→ 0] ← X: asmuo.
X[pagr_npd → 290] ← X: asmuo.
X[inpd → 430] ← X: asmuo [invalidumas → grupė(1); inpd → NPD], NPD < 430.
X[inpd → 380] ← X: asmuo [invalidumas → grupė(2)]; inpd → NPD], NPD < 380.
X[inpd → 430+(N-3)*46] ← X: asmuo [vaikų_skaičius → N; inpd → NPD], N ≥ 3, NPD < 430+(N-
3)*46.
X[inpd → 335+(N-1)*53] ← X: asmuo [vaikų_skaičius → N; ar_vienišas → taip; inpd → NPD];
N ≥ 1; NPD < 335+(N-1)*53.
X[inpd → 330] ← X: asmuo [ar_žemės_ūkio_darbuotojas → taip; inpd → NPD], NPD < 330.
X[pnpd → N*PNP*0.1/2] ← X: asmuo [vaikų_skaičius → N, pagr_npd → NPD; ar_vienišas →
ne], N ≥ 1, N <= 2.

```

1 lentelė

Vaikų skaičius	Ar vienišas	Invalidumas	Ar žemės ūkio darbuotojas	NPD	PNPD	Iš viso
0	ne	I gr.	taip/ne	430		430
1	ne	I gr.	taip/ne	430	14,50	444,50
2	ne	I gr.	taip/ne	430	29	459
3	ne	I gr.	taip/ne	430		430
4	ne	I gr.	taip/ne	476		476
5	ne	I gr.	taip/ne	522		522

X[bendras-*npd* → INPD+PNPD] ←-X: *asmuo* [*pagr_npd* → NPD; *inpd* → INPD; *pnpd* → PNPDP], *inpd*>0.

X[bendras-*npd* → NPD+PNPD] ←-X: *asmuo* [*pagr_npd* → NPD; *inpd* → INPD; *pnpd* → PNPDP], *inpd*=0.

}

Užklausa apie bendrą neapmokestinamą pajamų dydį priklausomai nuo tam tikrų asmens savybių (vaikų skaičiaus, invalidumo, šeiminių padėties, darbo žemės ūkyje) bus:

```
?-test module(s20) class(asmuo) methods(vaiku_skaicius{0, 1, 2, 3, 4, 5},
invalidumas, ar_vienišas, ar_zemes_ukio_darbuotojas) result-method(bendras-npd)).
```

1 lentelėje pateikta dalis rezultatų, iš kurių galima pastebėti, kad pirmos grupės invalidams, turintiems tris vaikus, bendras neapmokestinamas pajamų dydis yra mažesnis, negu turintiems du vaikus. Turintiems keturis vaikus šis dydis vėl padidėja. Aki vaizdu, kad tokio svyravimo neturėtų būti. Kitaip tariant, neapmokestinamą pajamų dydį apskaičiuojanti funkcija yra nemonotoniška pagal argumentą, nurodantį vaikų skaičių.

Norint automatizuoti nedarnos paiešką (pritaikyti antrąjį metodo lygmenį), užklausa papildoma dėmeniu *constraint(nedarna)*. Pats ribojimas formalizuojamas taip:

```
nedarna [tipas →> „NPD nemonotoniškumas pagal vaikų skaičių“] ←-
```

```
X1: asmuo [vaiku_skaicius → N1; invalidumas → Inv; ar_vienišas → V; ar_zemes_ukio_darbuotojas → U; bendras_npd → NPD1],
```

```
X2: asmuo [vaiku_skaicius → N2; invalidumas → Inv; ar_vienišas → V; ar_zemes_ukio_darbuotojas → U; bendras_npd → NPD2], N1>N2, NPD1 ≤ NPD2.
```

Analogiškai patikrinus NPD monotoniškumą pagal invalidumo grupę bus nustatyta, kad jis nėra pažeistas.

Išvados

1. Dėl įvairių kriterijų ir išimčių persipynimo teisės aktuose yra nemaža nedarnos tikimybė. Siūlomas metodas įgalina iš dalies automatizuotai tikrinti kiekybiškai išreikštų reikšmių darną.
2. Situacinio taikymo analizę ir ribojimų pažeidimų tikrinimą galima pritaikyti ne tik atskirų straipsnių analizei, bet ir, pavyzdžiui, globaliam kriminalinių bausmių balansavimui atsižvelgiant į vertybių klasifikatorių. Metodas turi būti modifikuojamas priklausomai nuo teisės aktų straipsnių tipų ir tikrinamos darnos ypatumų.

Literatūra

1. W. Voermans, Computer-assisted legislative drafting in the Netherlands: the LEDA-system, in: *Proceedings of National Conference on Legislative Drafting in the Global Village*, 16–17 November, Ottawa, Canada (2000).
2. A. Gonzalez, V. Barr, Validation and verification of intelligent systems – what are they and how are they different? *Journal of Experimental and Theoretical Artificial Intelligence*, **12**(4), 407–420 (2000).
3. S. Spreeuwenberg, R. Gerits, Requirements for successful verification in practice, in: *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference FLAIRS-2002*, AAAI Press (2002), pp. 221–225.
4. J.A. Wentworth, R. Knaus, H. Aougab, *Verification, Validation and Evaluation of Expert Systems*. An FHWA Handbook, Turner-Fairbank Highway Research Center, McLean, VA (1995).
<<http://www.tfhrc.gov/advanc/vve/cover.htm>> [žiūrėta 2004-11-12].
5. L. Paliulionienė, Teisinių žinių bazių vertinimo metodas, *Informacijos mokslai*, **32**, 110–117 (2005).
6. A. Čapliuskas, A. Dapšys, J. Misiūnas, V. Poškevičius, Kriminalinių bausmių sistemos daroma ir sankcijų optimizavimas kaip baudžiamosios politikos veiksmingumo prielaidos, *Teisė*, **37**, 7–15 (2000).
7. M. Kifer, G. Lausen, J. Wu, Logical foundations of object-oriented and frame-based languages. *Journal of the Association for Computing Machinery*, **42**(4), 741–843 (1995).
8. L. Paliulionienė, Ontologijos teisinių dokumentų rengimo ir analizės sistemos, *Informacijos mokslai*, **26**, 176–179 (2003).
9. T.R. Gruber, Towards principles for the design of ontologies used for knowledge sharing, *International Journal of Human-Computer Studies*, **43**, 907–928 (1995).

SUMMARY

L. Paliulionienė. On a method of legal knowledge base situational analysis

Many existing knowledge base verification and validation methods can be applied to legal knowledge bases. However, peculiarities of legal knowledge raise the need for specific methods. Internal and external consistency of legal acts is a very important element of their quality. One of the consistency types is the consistency of quantitative values, for example, criminal and administrative sanctions (penalty size, duration of imprisonment, etc.), fees, grants, and pensions. This paper presents a method for the analysis of quantitative results of possible situations in order to detect inconsistencies in the results.

Keywords: legal knowledge bases, legimatics, verification and validation, knowledge base consistency.