

KOMPIUTERINIS TEISINIŲ SAMPROTAVIMŲ MODELIAVIMAS

Laima Paliulionienė

Matematikos ir informatikos institutas, Akademijos g. 4, 2600 Vilnius
Telefonas 62 61 07
Elektroninis paštas laipal@ktl.mii.lt.

Pateikta 2000 m. sausio 4 d.

Parengta spausdinti 2000 m. gegužės 22 d.

Recenzavo Teisės instituto Informacinio metodinio skyriaus vyr. mokslinis bendradarbis doc. dr. V. Poškevičius ir Lietuvos teisės akademijos Valstybinio valdymo fakulteto Teisinės informatikos katedros vedėjas doc. dr. R. Petrauskas

S a n t r a u k a

Teisė yra viena iš tinkamiausių probleminių sričių dirbtinio intelekto metodams taikyti. Pirmos idėjos šioje srityje buvo iškeltos dar 1949 metais. Atliekant tyrimus kuriamos teisininkams naudingos kompiuterinės sistemos ir formalios metodikos. Kita vertus, teisė yra ideali analizės, tikrinimo ir eksperimentų sritis, kurioje galima kurti ir tobulinti pačius dirbtinio intelekto metodus.

Straipsnyje aptariama, kokie teisinių žinių ypatumai nulemia intelektualizuotų teisinių sistemų savybes ir kokiems uždaviniams spręsti šios sistemos naudojamos. Atsižvelgiant į uždavinius, išskiriami keli intelektualizuotų sistemų tipai – teisinės analizės, veiklos planavimo, koncepcinės informacijos paieškos ir teisės inžinerijos sistemos. Kitas sistemų klasifikavimo būdas – pagal modeliuojamų teisinių samprotavimų tradicijas. Čia išskiriamos statutinės teisės ir precedentinės teisės sistemos.

Straipsnyje nagrinėjama, kokie kompiuterinio modeliavimo metodai taikomi teisiniams samprotavimams modeliuoti. Apžvelgiami pagrindiniai metodai – dedukcija ir precedentinis samprotavimas, naudojantis indukcija ir analogija. Dažniausiai dedukcija taikoma teisinėms normoms ir euristikoms modeliuoti. Tai yra pagrindinis metodas statutinės teisės šalyse, prie kurių priskiriama ir Lietuva. Indukcija taikoma konkreitiems atvejams apibendrinti ir tokiu būdu gauti dedukcijos taisyklės. Analogija taikoma tada, kai norima priimti sprendimus atsižvelgiant į buvusius precedentes (sprendimus, priimtus panašiais atvejais). Sistemose, modeliuojančiose teisinius samprotavimus, šie metodai dažniausiai yra derinami vienas su kitu ir taip gaunami hibridiniai metodai. Straipsnyje aptariami ir kai kurie kiti šioje srityje taikomi metodai – argumentavimas, deontinė logika, hipertekstas, neuroniniai tinklai. Be to, atskirai nagrinėjami klausimai, susiję su kompiuteriniu modeliavimu rengiant teisinius dokumentus.

Socialiniai teisės aspektai riboja teisinių samprotavimų automatizavimo galimybes, todėl kompiuterinio modeliavimo eksperimentai sulaukia kai kurių teisininkų kritikos. Tačiau pripažįstama, kad tam tikru mastu kompiuterinis teisinių samprotavimų modeliavimas yra įmanomas ir gali padėti teisininkams priimant sprendimus.

1. Įvadas

Teisė yra viena iš tinkamiausių probleminių sričių dirbtinio intelekto metodams taikyti. Pirmi bandymai automatizuoti teisinius samprotavimus buvo susieti su teisinių ekspertinių sistemų kūrimu. Tokių sistemų idėja buvo iškelta dar 1949 metais [1]. Kai kurias idėjas vėliau taip pat pasiūlė L. Mehlas [2, p. 755], tačiau tik maždaug nuo 1970 m. kai kuriuose moksliniuose institutuose buvo pradėti aktyvesni tyrimai. Jų rezultatai iš pradžių buvo daugiausia teoriniai, sukurta tik keletas prototipinių sistemų. Nuo 1980 m. vykdoma jau gerokai daugiau tiriamųjų projektų, finansuojamų iš valstybinio bei privataus sektoriaus, sukurta daug prototipinių ir keletas komercinių sistemų. 1987 m. Bostone (JAV) įvyko pirmoji dirbtinio

intelekto ir teisės konferencija. 1992 m. pradėtas leisti tarptautinis žurnalas „*Artificial Intelligence and Law*“ (leidykla „*Kluwer Academic Publishers*“, Olandija).

Visų intelektualizuotų teisinių sistemų specifines savybes visų pirma nulemia teisinių žinių ypatumai:

1. Teisinės sąvokos dažnai būna nevienareikšmės, netikslios arba iš viso neapibrėžtos. Nevienareikšmiškumas gali būti semantinis (pavyzdžiui, „didelė žala“, „savanaudiški tikslai“, „trumpalaikis sveikatos sutrikimas“) arba sintaksinis. Semantinis nevienareikšmiškumas, neapibrėžtumai (*open texture*) sukelia konkrečių situacijų klasifikavimo problemą. Įstatymų rengėjai abstrakčias sąvokas dažnai vartoja sąmoningai, nes neįmanoma iš anksto numatyti visų galimų situacijų.
2. Teisiniai dokumentai yra struktūrizuoti, todėl paprastai keliamas izomorfizmo tarp įstatymo teksto ir jo formalizavimo reikalavimas.
3. Teisinės žinios yra nemonotoninės, turi gausybę išimčių. Jos taip pat turi metalygius ir deontinius modalumus.
4. Teisiniai dokumentai dažnai turi spragų ir nesuderinamumų. Kai kurie iš jų gali būti atsitiktiniai ir atsirasti dėl naujų prielaidų arba pataisų. Tačiau neišsamumas ir miglotumas gali atsirasti ir dėl sąmoningų įstatymų leidėjų ketinimų praktiniais ar politiniais tikslais, pavyzdžiui, siekiant palikti erdvę patobulinti greitai besikeičiančiose gyvenimo srityse, arba tiesiog todėl, kad nepavyko prieiti prie vienos nuomonės ir buvo reikalingas kompromisas.

Atsižvelgiant į tikslus, galima išskirti keturis pagrindinius intelektualizuotų sistemų tipus [3; 4, p. 413–421]. *Teisinės analizės sistemos* naudojamos konkrečios situacijos teisinėms pasekmėms įvertinti. Kai kuriais atvejais net įmanoma įstatymus taikyti automatiškai – pavyzdžiui, apskaičiuojant mokesčius. *Veiklos planavimo sistemų* paskirtis yra surasti geriausią veikimo būdą tam tikroje teisinėje aplinkoje – pavyzdžiui, siekiant minimizuoti mokesčius. *Koncepcinės informacijos paieškos sistemos* naudojamos teisinių dokumentų koncepcinės struktūros; informacija ieškoma, sutapatinant užklauso ir tekstų duomenų bazėje koncepcines struktūras, o ne žodžių grupes, kaip įprastose paieškos sistemose. *Teisės inžinerijos sistemos* skirtos padėti kurti įstatymus, siekiant sumažinti jų nesuderinamumo ir neišsamumo laipsnį.

Kitas sistemų klasifikavimo būdas – pagal modeliuojamų teisinių samprotavimų tradicijas. *Statutinės teisės* sistemų žinių bazėje saugomos taisyklės, gautos iš rašytinių įstatymų ir kitų teisinių dokumentų, ir paprastai naudojamas deduktyvinis išvedimas situacijos teisinėms pasekmėms įvertinti. Statutinė teisė naudojama daugelyje šalių, tarp jų ir Lietuvoje. *Precedentinės teisės* sistemų žinių bazėje saugomos ankstesnių bylų situacijos bei jų įvertinimai. Jose naudojamas induktyvinis ir analoginis išvedimas naujos bylos ir senų bylų faktams palyginti, panašioms byloms ieškoti ir analogiškiems sprendimams konstruoti. Precedentinė teisė naudojama daugelyje JAV valstijų, Anglijoje ir kai kuriose kitose šalyse. Dažnai vienoje sistemoje modeliuojami abu teisinio samprotavimo būdai. To priežastis yra neišvengiamas teisinių sąvokų neišsamumas net šalyse, kuriose gyvos statutinės teisės tradicijos: galutinį sprendimą dėl įstatymų taikymo konkrečiu atveju gali priimti tik teismas.

Šio straipsnio tikslas yra apžvelgti kompiuterinio samprotavimų modeliavimo metodus, taikomus dirbtinio intelekto teorijoje, ir tų metodų taikymą teisiniams samprotavimams modeliuoti. Apžvelgiami pagrindiniai samprotavimų modeliavimo metodai – dedukcija, indukcija ir analogija, taip pat kai kurie kiti metodai, taikomi modeliuojant teisinius samprotavimus – argumentavimas, deontinė logika, hipertekstas, neuroniniai tinklai. Be to, atskirai nagrinėjami klausimai, susiję su kompiuteriniu modeliavimu rengiant teisinius dokumentus.

2. Dedukcijos metodas

Dedukcija vadinamas samprotavimas nuo bendro link konkrečiau. Taikant dedukcijos metodą, imamos bendrus teiginius modeliuojančios formulės, vadinamos *prielaidomis*, ir, taikant *loginio išvedimo taisykles*, žingsnis po žingsnio konstruojamos konkrečius tvirtinimus modeliuojančios formulės, vadinamos *išvadomis*.

Loginio išvedimo taisyklės išsaugo tvirtinimų teisingumą, todėl iš teisingų prielaidų visuomet yra gaunamos teisingos išvados. Dedukcijos žingsnių visuma vadinama *formaliu įrodymu*. Taigi, taikant šį metodą, samprotavimai yra modeliuojami formaliu įrodymu.

Dedukcijos metodas pasiūlytas dar antikos laikais. Pirmasis jį taikė Aristotelis (IV a. pr. m. e.). Jis įrodinėjo tvirtinimus naudodamas vadinamuosius *silogizmus*, kuriuos sudarė dvi prielaidos ir išvada, pavyzdžiui:

Visi žmonės mirtingi
Sokratas yra žmogus
Vadinasi, Sokratas yra mirtingas.

G. Frege'as 1879 m. pirmasis pasiūlė griežtai aksiomatizuotą predikatų skaičiavimą. Tai buvo formali žymėjimų sistema, pritaikyta deduktyviems samprotavimams atlikti mechaniškai. Pavyzdžiui, naudojant predikatus, teiginį "Sutartis yra galiojanti, jei ji parašyta raštu, pasirašyta ir patvirtinta notaro" galima užrašyti taip:

galiojanti_sutartis (X) ← sutartis (X), parašyta_raštu (X), pasirašyta (X), patvirtinta (X, Y), notaras (Y).

Pirmoji kompiuterinė teoremų įrodymo sistema buvo 1954 m. M. Daviso sukurta vadinamoji Presburgerio įrodymo programa, kuri gebėjo įrodinėti kai kurias adityviosios aritmetikos teoremas [5]. Pirmosiose sistemose buvo realizuotos gana neveiksmingos įrodymo procedūros, tačiau jos buvo nuolat tobulinamos. Esminis rezultatas buvo J. Robinsono pasiūlytas rezoliucijos principas [6, p. 23–41]. Naudojant įvairius šio principo variantus (patikslinimus), buvo sukurta nemažai labai veiksmingų sistemų. Vienas iš plačiausiai žinomų praktinių rezultatų yra programavimo kalba Prolog. Rezoliucijos principas taikomas daugumoje vėliau sukurtų deduktyvinių samprotavimų sistemų.

Daugelyje dalykinių sričių susiduriama su *nemonotoniniais samprotavimais*. Nemonotoniniais vadinami tokie samprotavimai, kai tvirtinamas kartą pripažintas teisingu, vėliau gali būti paneigtas. Teisėje tokie samprotavimai vaidina ypač svarbų vaidmenį, nes visuose teisiniuose aktuose gausu išimčių. Darbas su nemonotoniniais samprotavimais prasidėjo nuo R. Reiterio darbų duomenų bazių srityje [7, p. 55–77]. Svarbus yra L. T. McCarty darbas [8, p. 304–311], kuriame parodyta, kad nors teoriškai išimtis galima išreikštiniu būdu įtraukti į taisyklių prielaidas (pavyzdžiui, *skraido (X) ← paukštis (X), not pingvinas (X)*), tačiau praktiškai toks sąrašas būtų begalinis, ir yra labai neefektyvu keisti taisykles kiekvieną kartą, atsiradus naujoms išimtims. Vėliau buvo sukurta įvairių darbo su išimtimis metodų – apėmimas, nutylėjimų logikos, modalinės logikos [9, p. 95–126].

Teisiniams samprotavimams modeliuoti dedukcijos metodas pradėtas taikyti pirmasis. Jis neblogai tinka, kai teisiniai samprotavimai grindžiami įstatymu (statutinė teisė), t.y. kai neatsižvelgiama į precedentus, paprotinę teisę ir kitus panašaus pobūdžio veiksnius. Taip yra vadinamosiose civilinės (romėnų) teisės šalyse, prie kurių priskiriama ir Lietuva. Teisinių ekspertinių sistemų, taikančių dedukcijos metodą, idėja buvo iškelta dar 1949 metais [1]. Teisininkas L. Allenas, 1957 m. pasiūlęs naudoti formalizuotą logikos kalbą rengiant ir analizuojant teisinius dokumentus [10, p. 833–879], taip pat kalbėjo apie dedukcijos metodu mechaniškai išvedamas pasekmes.

Viena iš pirmųjų intelektualizuotų teisinių sistemų buvo sistema TAXMAN [11, p. 837–893], skirta korporacinių mokesčių sričiai. Ji buvo realizuota kalboje micro-PLANNER. Sistemoje TAXMAN žinios vaizduojamos semantiniais tinklais, kuriuos galima lengvai transformuoti į predikatų logiką, ir naudojamas deduktyvinis išvedimas. Vėliau, sistemoje TAXMAN II, L. T. McCarty įvedė ir analoginius samprotavimus [8, p. 304–311].

Dažniausiai cituojamas kompiuterinio teisės modeliavimo darbas yra Britanijos pilietybės įstatymo formalizavimas [12, p. 370–386]. Šis įstatymas buvo užrašytas kaip loginė programa, naudojant ekspertinių sistemų kūrimo sistemą APES, kuri yra išplėstas Prologo variantas. Štai vieno straipsnio formalizavimo pavyzdys:

1.-(1) A person born in the United Kingdom after commencement shall be a	<i>X acquires British citizenship on date Y by sect. 1.1</i>
--	--

<p>British citizen if at the time of birth his father or mother is</p> <p>(a) a British citizen; or</p> <p>(b) settled in the United Kingdom.</p>	<p><i>if X was born in the U.K. and X was born on date Y and Y is after or on commencement and X has a parent who qualifies under 1.1 on date Y</i></p>
---	---

Programos išvedimo rezultatai yra loginės išvados iš įstatymą formalizuojančių taisyklių ir informacijos, gautos dialogo su naudotoju būdu. Programa taip pat pateikia išvedimo paaiškinimus. Pagrindinė problema autoriams buvo darbas su išimtimis (neigiama informacija). Ši problema buvo sprendžiama trejopai – neigimas kaip nesėkmė (not A išvedamas, jei nepavyksta išvesti A), neigiamos informacijos formalizavimas kaip teigiamos bei neigiamos informacijos įvedimas dialoge su naudotoju.

Britanijos pilietybės įstatymo formalizavimo darbas prasidėjo nuo M. J. Sergot tyrimų, susijusių su R. Stamperio kalba LEGOL [13, p. 293–305] ir jos ryšiu su loginiu programavimu. LEGOL tikslas buvo sukurti koncepcinio modeliavimo metodus organizacijų informacinėms sistemoms kurti. Daugelis tokių sistemų grindžiama tam tikrais įstatymais arba vidinėmis taisyklėmis, todėl teisinių dokumentų vaizdavimas kompiuteryje buvo vienas iš LEGOL tikslų. Daugiausia dėmesio buvo skirta teisės semantinio modelio sukūrimui. Šis modelis tapo pagrindu programavimo kalbai, kurioje galima užrašyti taisykles ir imituoti teisinių sąlygų pasekmes. Įstatymo fragmentas transformuojamas į LEGOL sakinius dviem žingsniais: (1) įstatymo sąvokos ir sąryšiai analizuojami ir identifikuojami pagal LEGOL semantinį modelį, (2) naudojant tradicines procedūrinio programavimo valdymo struktūras (seką, alternatyvą, ciklą) ir identifiкуotas sąvokas, užrašomos taisyklės.

Sistemose dažniausiai vienaip ar kitaip naudojamas nemonotoninis išvedimas, kadangi tokia yra teisinių žinių prigimtis. Formalizuojant Britanijos pilietybės kodeksą, nemonotoniškumas yra dėl uždaro pasaulio prielaidos, taikomos Prologe. Dalykinės srities išimtys čia įtraukiamos į bendrų taisyklių formulavimą. Išreikštinis išimčių vaizdavimas greta pagrindinių taisyklių ir nemonotoninis išvedimas naudojami, pavyzdžiui, sistemoje OBLOG [14, p. 231–239].

Kai kuriose sistemose dedukcija praplečiama neraiškiosios logikos (*fuzzy logic*) elementais [15]. Neraiškiosios aibės sąvoką 1965 m. įvedė Zadehas, kad galima būtų formaliai kalbėti apie tokias netikslias sąvokas kaip, pavyzdžiui, “didelis” ir “mažas”, ir atsakyti ne tik “taip” arba “ne”, bet ir nurodyti artumo laipsnį intervale [0; 1]. Tokio pat pobūdžio yra semantinis neapibrėžtumas teisėje. Tačiau kol kas neraiškiosios logikos panaudojimas teisėje sulaukė daugiau kritikos negu pritarimo, nes “jos filosofinis pagrindas yra abejotinas, taikant ją teisinėms sąvokoms” [16, p. 49].

Atspindint normatyvinę teisės prigimtį kartais naudojami deontiniai operatoriai “privalama”, “draudžiama”, “leidžiama” ir kt. Pavyzdžiui, sistemoje Prodeon [17, p. 31–43] teisinės žinios vaizduojamos pagal L. Alleno pasiūlymus, naudojant elementarius natūralios kalbos sakinius, sujungtus loginėmis jungtimis, IF-THEN taisykles, taip pat kintamuosius ir konstantas sakinių viduje bei deontinius operatorius. Sistemoje galimas tiesioginis išvedimas visoms išvadoms gauti bei atvirkštinis išvedimas atsakant į užklausą. Detaliau deontinės logikos elementų taikymą aptarsime vėliau.

Teisinių žinių metalygiams atspindėti kartais naudojamas loginis metaprogramavimas. Pavyzdžiui, [18, p. 55–80] aprašomas loginio metaprogramavimo kalba Alloy (išplėstas Prologo variantas), kurioje galima operuoti teorijomis (programomis) ir metateorijomis (metaprogramomis) aprašant teisines taisykles ir metataisykles.

Dedukcija taip pat naudojama sistemose, kuriose pabrėžiamas ne įstatymų modeliavimas, o sprendimų priėmimo procesų modeliavimas, atsižvelgiant į įstatymuose nenumatytus faktorius, kurie gali lemti bylos nagrinėjimo rezultatą. D. A. Watermano ir M. A. Petersono sistema LDS (*Legal Decision-making System*) [19, p. 272–275], realizuota kalboje ROSIE (*Rule-Oriented System for Implementing Expertise*), pataria teisininkui, ar verta imtis bylos dėl atsakomybės už gaminio sugadinimą, atsižvelgiant į tokius faktorius kaip teisininkų galimybės, abiejų šalių charakteristikos, laikas, nuostolio pobūdis, tikimybė nustatyti atsa-

kingajį ir pan. Panašaus pobūdžio yra tų pačių autorių 1986 m. sukurta sistema SAL (*System for Asbestos Litigation*). Australijos valstybinio draudimo departamente buvo sukurta sudėtinga sistema COLOSSUS, skirta atpažinti potencialią apgaulę asmens sužalojimo bylose ir atitinkamai pažymėti jas, kad inspektorius papildomai patikrintų [20, p. 677–680]. Šios sistemos priskirtinos prie veiklos planavimo sistemų, kaip ir sistema TAXADVISOR, sukurta R. H. Michaelsono EMYCIN terpėje [21, p. 149–167]. Ji skirta pasiūlyti tokį didelio turto reorganizavimą, kuris minimizuotų mokesčius.

Dažniausiai dedukcija naudojama teisinėms normoms ir euristikoms modeliuoti. Tačiau yra pasiūlyti keli būdai, kaip į taisyklinę sistemą įtraukti žinias apie precedentes. J. C. Smithas ir C. Deedmanas [22, p. 84–93] tam naudoja giluminės struktūros (*deep structure*) sąvoką, kuri pas juos reiškia faktų šablonus. Ekspertai teisininkai nustato šias struktūras iš precedentų konkrečioje probleminėje srityje ir iš jų formuoja taisyklės. Kitą būdą pasiūlė T. Bench-Caponas [23, p. 37–45]. Jis teigia, kad tam tikrose srityse (pavyzdžiui, mokesčių įstatymų srityje) neaiškios bylos atveju sukuriama nauja norma. Todėl taisyklės, gautos iš įstatymų, reikia papildyti taisyklėmis, kurias pateikia asmuo, priimantis sprendimus šioje srityje.

Dedukcijos metodo taikymas teisiniams samprotavimams ir ekspertinių sistemų kūrimas yra sulaukęs ir kritikos, ypač iš teisininkų [15]. Paminėsime kai kurias problemas:

1. Teisė nėra vien tik įstatymų tekstas, ji yra neatskiriama nuo socialinio ir politinio konteksto, todėl kai kurie teisininkai teigia, kad neiškreipus teisės neįmanoma jos formalizuoti taip, kad tiktų formaliam deduktyviam išvedimui, nebent labai siaurose ir nereikšmingose srityse.
2. Kai kurie teisės filosofai neigia dedukcijos naudą, teigdami, kad teisėjai pradeda nuo intuityviai gautos išvados ir pagal ją parenka prielaidas ir taisykles, t.y. faktiškai naudoja ne dedukciją, o abdukciją.
3. Egzistuoja darbo su semantiniu neapibrėžtumu problema. Deduktyviniai samprotavimai nesugeba klasifikuoti konkrečių situacijų pagal negriežtai apibrėžtas sąvokas ir paprastai naudotojas turi tai padaryti pats. Viena iš induktyvinių ir analoginių samprotavimų stiprioji pusė kaip tik ir yra tai, kad jie gali padėti spręsti negriežtų sąvokų problemą, atsižvelgiant į sąvokos traktuotę ankstesniuose precedentuose.

3. Indukcijos ir analogijos metodai. Precedentinis samprotavimas

Indukcija vadinamas samprotavimas nuo konkretaus link bendro, iš atskirų atvejų generuojant bendras taisykles, pavyzdžiui:

Adomas yra žmogus, ir jis mirtingas
Mozė yra žmogus, ir jis mirtingas
Sokratas yra žmogus, ir jis mirtingas
Napoleonas yra žmogus, ir jis mirtingas
Džordžas Vašingtonas yra žmogus, ir jis mirtingas
Vadinasi, visi žmonės mirtingi

Pirmieji esminiai rezultatai kaip modeliuoti samprotavimus, naudojant automatinę indukciją, buvo gauti septintojo dešimtmečio viduryje. Indukcija grindžiamose sistemose taikomi įvairūs metodai hipotezėms generuoti ir tikslinti pagal pateikiamus pavyzdžius.

Analogijos metodu vadinamas toks samprotavimas, kai pagal vienus atskirus atvejus daromos išvados apie kitą panašų į juos atvejį, t.y. samprotavimas nuo konkretaus link kito konkretaus, pavyzdžiui:

X turi savybes F, G, ...
Y turi savybes F, G, ...
Be to, X turi savybę H
Vadinasi, Y taip pat turi savybę H

Analogijos metodas dažnai laikomas indukcijos atmaina, kadangi ir vienu, ir kitu atveju samprotavimas prasideda nuo konkrečių atvejų analizės. Skirtingai nuo dedukcijos, kurioje teisingos prielaidos garantuoja teisingas išvadas, indukcijoje ir analogijoje teisingos prielaidos tik tam tikru laipsniu padidina tikimybę, kad išvados yra teisingos, bet to negarantuoja. Pavyzdžiui, matydami tik skraidančius paukščius, galėtume padaryti išvadą, kad visi paukščiai skraido, tačiau tai nebūtų tiesa.

Palyginti nauja dirbtinio intelekto paradigma yra *precedentinis samprotavimas (case-based reasoning)*, kuriame naujos problemos sprendžiamos kaupiant, ieškant ir priderinant ankstesnių problemų sprendimus. Precedentinio samprotavimo sistemos paprastai apima precedentų bazę, tam tikrą jų saugojimo metodą, jų indeksavimo schemą, dalinio sugretinimo metodą, siekiant surasti tinkamiausius precedentus, bei senų sprendimų pritaikymo naujai problemai metodą.

Pirmos precedentinio samprotavimo sistemos atsirado devintojo dešimtmečio antrojoje pusėje. Jos naudojamos teisės, medicinos, projektavimo, planavimo ir kitose srityse. Precedentinis samprotavimas turi daug bendro su indukcijos ir analogijos metodais ir naudoja jų elementus. Indukcija dažnai taikoma ir indeksuojant precedentų bazę greitesnei precedentų paieškai – tam naudojamas induktyviai iš pavyzdžių sukonstruotas sprendimų medis.

Precedentinis samprotavimas yra pagrindinis samprotavimo metodas precedentinės teisės šalyse, ir būtent šių šalių mokslininkai labiausiai jį išplėtojo. Be to, tai yra būdas spręsti semantinio neapibrėžtumo problemą deduktyviniame samprotavime – kai įstatyme užrašyta norma nėra akivaizdžiai tinkanti konkrečiam atvejui, galima pritaikyti apibendrinimą iš ankstesnių precedentų.

Tam, kad galima būtų įvertinti bylą panašumą, situacijos paprastai vaizduojamos kaip freimai, kurių slotai aprašo esmines bylos savybes. Taikant tam tikrą statistinį metodą, šioms savybėms gali būti priskiriami svoriai. Svoriai gali būti priskiriami ir pačioms byloms – pavyzdžiui, pagal bylą sprendusio teismo lygį. Problemos dažnai atsiranda dėl to, kad sunku nustatyti, kokios bylos savybės gali būti svarbios ateityje. Gali būti keli precedentinių samprotavimų lygiai: (1) intelektualizuota panašių precedentų paieška (paprasčiausias atvejis), (2) išvadų apie galimą sprendimą pateikimas, (3) argumentų konstravimas norimam sprendimui pasiekti. Argumentavimo metodas plačiau apžvelgtas kitame skyriuje.

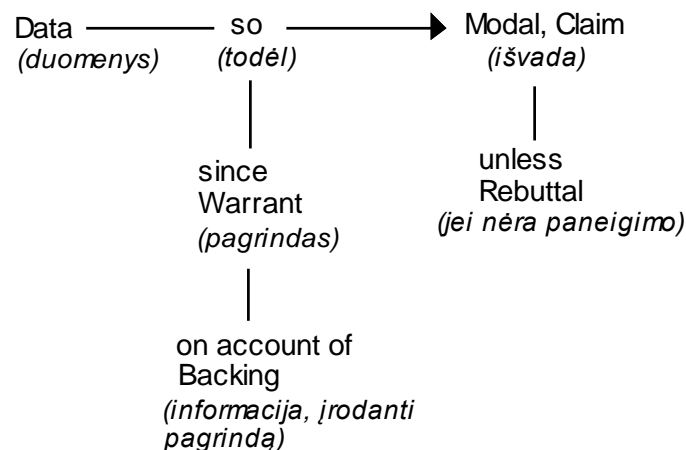
Klasikiniai teisinių precedentinių samprotavimų sistemų pavyzdžiai yra TAXMAN II ir HYPO. TAXMAN II yra projekto TAXMAN tęsinys, kuriame pasiūlyta vadinamoji “prototipai plus deformacijos” struktūra semantiniams neapibrėžtumams spręsti analogijos metodu [8, p. 304–311]. Joje negriežtoms sąvokoms apibrėžti naudojami trys komponentai: (1) invariantas, aprašantis būtinas, bet dar nepakankamas sąvokos galiojimo sąlygas, (2) prototipų-pavyzdžių aibė, demonstruojanti situacijas, kuriose sąvoka galioja, (3) transformacijų aibė, aprašanti galimus atvaizdavimus iš prototipą į prototipą ir iš naujos situacijos į prototipą. Kadangi nauja situacija gali būti transformuota į skirtingus prototipus, ja kvalifikuojant generuojami alternatyvūs argumentai, vedantys prie skirtingų sprendimų.

Sistema HYPO pateikia trijų lygių abstrakcijas byloms aprašyti [24] – freiminę bylų ir situacijų vaizdavimo kalbą, faktinius predikatus ir matmenis. Faktiniai predikatai aprašo tam tikrus teisinius teiginius apie bylą. Jie naudojami kaip sąlygos nustatyti bylos matmenis, pagal kuriuos sistema nustato bylų panašumą (taikant dalinį sutvarkymą ir konstruojant gardele) ir naudoja argumentuoti. Sistema gali konstruoti hipotetines bylas argumentui patikslinti.

Kiti teisinių precedentinio samprotavimo sistemų pavyzdžiai yra SHYSTER [25], CABARET [26, p. 839–887].

4. Argumentavimo metodas

Nuo devintojo dešimtmečio pabaigos argumentavimas yra viena iš pagrindinių bendrų dirbtinio intelekto ir teisės tyrimų temų. *Argumento* schemą, kuri yra daugelio šių tyrimų pagrindas, dar 1958 m. pasiūlė S. Toulminas:



Šio samprotavimo metodo ypatumas yra tas, kad teiginys ne įrodomas įprasta prasme, o “pateisinamas” (*justified*). Argumentams pateikiami kontrargumentai ir t.t. Tokiame samprotavime gali nebūti vienintelės teisingos išvados. Argumentavimas modeliuoja teisės dialektiką bei teisininkų debatus. Šis metodas taikomas ir kitose tyrimų srityse, pavyzdžiui, kolaboratyviniame mokyme.

Argumentavimas nagrinėjamas skirtingais požiūriais. Viena kryptis yra “argumentavimo schemas” (AF – *argumentation framework*) – nemonotoninio argumentavimo sistemos, kitaip dar vadinamos “deklaratyviosiomis”, arba “reliacinėmis”. Dauguma AF pasiūlyta vykdant bendro pobūdžio nemonotoninių samprotavimų tyrimus. Juose taikoma idėja, kad nemonotoninis išvedimas gali būti analizuojamas per sąryšius tarp argumentų, grindžiančių alternatyvias išvadas. Teisinis samprotavimas taip pat yra nemonotoninis, ir todėl ne nuostabu, kad kelios argumentavimo schemas pritaikytos ir teisės srityje [27, p. 25–75; 28, p. 202–211]. Daugumoje tokių sistemų dirbama su normų prioritetais ir metaprogramavimu. Išskiriami penki struktūriniai AF elementai: (1) tam tikra logikos kalba, (2) argumento api-

brėžtis, (3) konflikto tarp argumentų apibrėžtis, (4) vieno argumento nurungimo (anuliavimo) kitu argumentu apibrėžtis, (5) argumentų įvertinimo metodas. Pavyzdžiui, H. Prakkeno ir S. Sartoro darbuose vertinami argumentai skirstomi į tris klases – tie, su kuriais galima laimėti debatus (*justified*), pralaimėti juos (*overruled*), ir tie, kurie neišsprendžia debatų (*defensible*). Argumentas čia yra konkretizuotų taisyklių seka, vedanti prie tam tikros išvados. Įrodymo teorija apibrėžiama kaip dialoginis žaidimas: įrodymas yra dialogas tarp argumento gynėjo ir oponento, kuriame kiekvienas žaidėjas bando nurungti kito žaidėjo ėjimą. Laikoma, kad argumentas laimi debatus, jei kaip oponentas beatakuotų, jo ėjimui išsenka.

Kita argumentavimo tyrimų kryptis yra procedūrinių debatų savybių nagrinėjimas [29; 30, p. 146–155]. Pagrindinė idėja yra ta, kad debatų efektyvumas, teisingumas ir racionalumas priklauso nuo procedūrinių aspektų, ir todėl reikia formalizuoti dialoginio žaidimo protokolą, įvedant taisykles, nusakančias dalyvių teises ir įsipareigojimus bei apibrėžiančias, kokie argumentai yra priimtini. Teisinis argumentas laikomas priimtiniu, jei jį galima apginti nuo kritikos teisingai vedamuose debatuose. Pagrindinis tokių tyrimų tikslas yra nustatyti bendrą dialektinių protokolų formą ir iširti, kokios savybės lemia šio protokolo efektyvumą.

Trečia kryptis – tai tyrimai precedentiniame samprotavime, kuriuose modeliuojami teisininkų argumentavimo būdai tikruose teismo procesuose [24; 26, p. 839–887]. Šios krypties darbai labiau susiję su retoriniais aspektais, teisininkų argumentavimo strategija bei taktika, o apie argumento korektiškumą logikos prasme nekalbama. Toks argumentavimas paprastai taikomas HYPO tipo sistemose. HYPO sistemoje [24] rasti precedentai naudojami teisinio argumento schemai suformuoti. Argumente cituojamos bylos už ir prieš tam tikrą sprendimą. Argumento struktūra priklauso nuo rastų precedentų ir jų tarpusavio sutvarkymo pagal panašumą į nagrinėjamą atvejį. Galimi ir papildomi kriterijai. Pavyzdžiui, darbe [31, p. 50–59] siūloma įtraukti į precedento aprašą ir naudoti argumentuojant ne vien tik faktus, bet ir teleologinį aspektą, t.y. teismo tikslą priimant nuosprendį (“siekiant apginti ieškovo gerovę”, “siekiant išvengti panašių bylų antplūdžio ateityje” ir pan.), ir įvesti keturių tipų santykius tarp tikslų: taksonominis, apribojimo, prieštaravimo ir instrumentinis (kai vienas tikslas padeda pasiekti kitą tikslą).

Taigi yra dvi viena kitą papildančios argumento sampratų – loginė, t.y. ryšys tarp prielaidų ir išvados, ir dinaminė, t.y. ėjimas debatuose. Gerą argumentavimo metodų apžvalgą galima rasti darbuose [32, p. 331–368; 33].

5. Deontinė logika

Deontinė logika yra modalinės logikos rūšis, kurioje modeliuojami normatyviniai samprotavimai, skiriant idealias ir realias situacijas ir naudojant modalinius operatorius “privaloma”, “leidžiama”, “draudžiama” [34, p. 17–40]. Galimi ne tik standartiniai deontiniai operatoriai, bet ir skirtingi jų variantai – “pareiga”, “teisė”, “įgaliojimas”, “atsakomybė” ir kiti. Žemiau pateiktas formalizavimo pavyzdys, kuriame naudojamas deontinis operatorius O (“obligatory” – privaloma). Čia p reiškia “knygos į biblioteką grąžintos laiku”, q – “naujos knygos neišduodamos”:

1. Knygas į biblioteką reikia grąžinti laiku.	1. Op
2. Jei knygos negrąžinamos laiku, naujos knygos neišduodamos.	2. $\sim p \rightarrow Oq$

Modalinės logikos, kaip ir visos logikos, šaknys yra Aristotelio darbuose, kuriuose jis pradėjo skirti realius faktus nuo būtinų ir galimų. Šiuolaikinė modalinė logika prasidėjo šio šimtmečio pradžioje, nuo C. I. Lewiso bandymų išspręsti materialiosios implikacijos problemas. Jis įvedė griežtą implikaciją \prec ir pasiūlė penkias skirtingas logines sistemas. Jo formulė $p \prec q$ suprantama kaip $O(p \supset q)$, kur O yra modalinis būtinumo operatorius. Nors panašiu laiku buvo pasiūlytos dar kelios modalinės logikos, sistemingiau jomis domėtis pradėta tik

septintajame dešimtmetyje. 1963 m. S. Kripke'as pasiūlė vėliau tapusią standartine modalinės logikos semantiką, vadinamąją *galimų pasaulių semantiką*. Pagrindinė jos idėja – tai, kas yra būtina, turi būti teisinga visuose galimuose pasauliuose, o tai, kas yra galima (įmanoma), turi būti teisinga bent viename iš šių pasaulių, nors realiame pasaulyje gali būti ir klaidinga.

Pirmą deontinę logiką 1926 m. sukūrė E. Mallis. 1951 m. G. H. von Wrightas pasiūlė vadinamąją standartinę deontinę logiką SDL. Deontinės logikos pradininkais taip pat buvo H. N. Castañeda ir C. E. Alchourrónas.

Deontinėse logikose susiduriama su problemomis, vadinamomis *paradoksis* [35], iš kurių labiausiai žinomas yra Chisholmo paradoksas. Jis susijęs su prieštaravimu tarp normų, standartinėje deontinėje logikoje išvedamų iš realios ir iš idealios situacijos, t.y. iš faktinio ir deontinio atskyrimo. Kitaip tariant, jis susijęs su normų pažeidimų apdorojimo normomis (*contrary-to-duty imperatives*).

Siekiant išspręsti paradoksus, buvo kuriamos vis naujos deontinės logikos – G. H. von Wrighto diadinė deontinė logika (1964, 1965 m.), B. C. van Fraasseno sąlyginė deontinė logika (1972 m.), B. Löwerio ir M. Belzerio sistema 3-D, kurioje įvedamas laikas (1983 m.), A. J. I. Joneso ir I. Pörno sistema DL (1985 m.), A. Soetemano diadinė sistema W-64 (1989 m.) ir t.t. Tačiau, sprendžiant vienas problemas likdavo arba atsirasdavo kitų. Kai kurios sistemos yra per sudėtingos, kai kurios gali dirbti tik su iš anksto numatytais išimtimis, daugelyje nėra fiksuojamas normos pažeidimas, kai yra to pažeidimo apdorojimo norma ir t.t.

Vienas iš pirmųjų deontinių operatorių taikymo teisėje pavyzdys yra R. Stamperio projektas LEGOL [13, p. 293–305], apie kurį kalbėjome skyriuje apie dedukciją. Vienas iš pasiūlytos kalbos išplėtimų turėjo operatorius darbui su deontinėmis sąvokomis “teisė”, “pareiga”, “privilegija” ir “atsakomybė”, tačiau jis taip ir nebuvo realizuotas.

Kitas pavyzdys – sistema ESPLEX [36, p. 240–252], skirta kurti ekspertines sistemas skirtingoms teisės sritims. Ji turi panašią į Prologą sintaksę ir naudoja deontinius operatorius, tačiau joje yra tik deontinės logikos užuomazgos.

Dar vienas pavyzdys – projekto TAXMAN išplėtimas. Jame L. T. McCarty pasiūlė savo diadinės deontinės logikos versiją. Be deontinių operatorių, buvo įvestos ir konstrukcijos veiksmams specifikuoti. Pridėjus tipizavimo, įvykių ir laiko konstrukcijas, buvo sukurta kalba LLD (*a Language for Legal Discourse*). Tačiau ši kalba nebuvo realizuota.

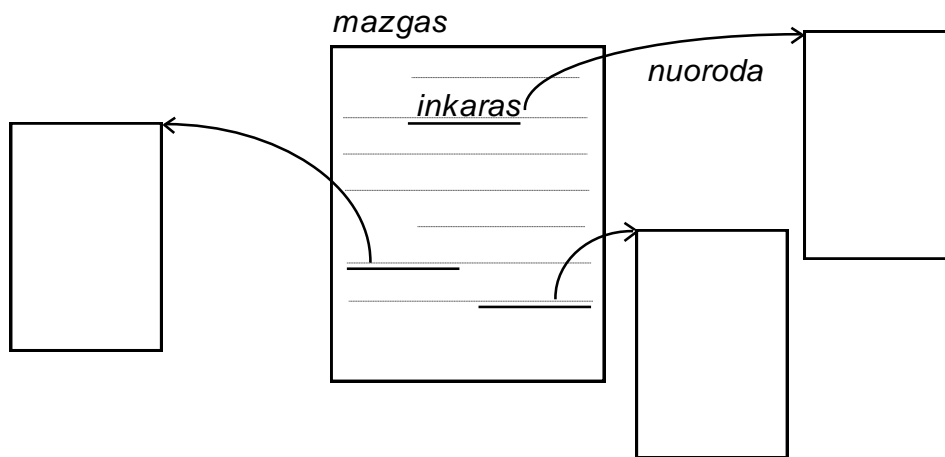
1985 m. L. Allenas ir Ch. Saxonas pasiūlė [37, p. 385–450] formalią kalbą teisinių tekstų analizei siekiant surasti juose dviprasmybių. Tam jie naudojo sąvokų sistemą, kurioje buvo ir skirtingi standartinių deontinių operatorių variantai – “leidimas”, “privalėjimas”, “draudimas”, “teisė”, “pareiga”, “privilegija”, “įgaliojimas”, “atsakomybė”, “neliečiamumas”. Jie sukūrė sistemą MINT, kuri galėjo generuoti skirtingas teksto interpretacijas ir taip padėti pamatyti nevienareikšmiškumą. Pavyzdžiui, Londono imperatoriaus koledžo bibliotekos taisyklėse buvo parodyta 2560 skirtingų interpretacijų. Panašiais tikslais deontiniai operatoriai naudojami ir Prodeone [38, p. 31–43].

Taigi galima išskirti trijų tipų deontinės logikos taikymus teisėje: teisės aktų rengimas (MINT), teisės aktų taikymas (ESPLEX) ir teisininkų mąstymo modeliavimas (TAXMAN išplėtimas). Pažymėtina, kad pastaruoju atveju formalizuojamas ne teisinis dokumentas, o teisininkų mąstymo psichologinis procesas. Deontinė logika naudojama ne tik teisėje, bet ir kompiuterinių sistemų, tinklų saugumo reikalavimams, duomenų bazių apribojimams specifikuoti ir kt. [34, p. 17–40].

Nors ir atrodo, kad deontinė logika tinkama teisei, ji šioje srityje nėra plačiai naudojama [15; 39; 40]. Tam yra dvi priežastys. Viena jų susijusi su pačia deontine logika: joje iki šiol lieka daug problemų ir ji dar niekur nėra iki galo realizuota. Kita priežastis yra tai, kad teisėje normatyvinė dalis nėra pagrindinė. Ji yra svarbi tik palyginti siaurose srityse, pavyzdžiui, bibliotekos taisyklėse. Todėl deontinės logikos naudojimas yra tik užuomazginis. Tačiau deontinė logika gali padėti formaliai nustatyti, ar teisinis dokumentas yra suderintas aprašomų pareigų ir teisių atžvilgiu, ir todėl gali būti naudinga rengiant įstatymus ir siekiant sumažinti jų nesuderinamumo ir neišsamumo laipsnį.

6. Hipertekstas

Hipertekstas yra netiesinio (asociatyviojo) žinių vaizdavimo ir valdymo metodų apibendrintas pavadinimas. Hipertekstinės sistemos imituoja žmogaus smegenų gebėjimą saugoti ir ieškoti informacijos pagal asociatyvius ryšius. Skirtingai nuo nuoseklių žinių valdymo metodų, kai paieškos erdvė yra sutvarkyta pagal poklasio arba sekos santykius (pavyzdžiui, pagal abėcėlę arba eilės tvarka), asociatyvieji žinių valdymo metodai naudoja orientuotą grafą, kuriame kiekvienas *mazgas* yra tam tikras informacinis vienetas. Mazgas sieja orientuoti ryšių lankai – *nuorodos*. Daugelyje hipertekstinių sistemų mazgas gali turėti kelias nuorodas į kitus mazgus, kurių kiekviena yra susieta su kuria nors mazgo dalimi, vadinama *inkaru*. Kai aktyvuojamas vienas iš inkarų, pagal su juo susietą nuorodą pereinama prie kito



mazgo. Pagal tą pačią nuorodą galima grįžti ir atgal (*backtracking*).

Pirmas asociatyviųjų ryšių idėją 1945 m. išsakė V. Bushas, kuris pasiūlė teorinį modelį “Memex”, skirtą didelei mikrofišų bibliotekai. Terminą “hipertekstas” septintajame dešimtmetyje įvedė T. Nelsonas.

Paprastai mazgas įsivaizduojamas kaip naudotojui pateikiamas ekrano vaizdas, kuriame gali būti tekstai, paveikslai, animuoti vaizdai, taip pat garsinė informacija (kitaiip dar vadinama *hipermedia* – hiperteksto ir *multimedia* junginys). Tačiau plačiąja prasme hipertekstas yra bendresnė sąvoka, ir mazgais iš tikrųjų gali būti, pavyzdžiui, antecedentai, konsekvantai, procedūros, o nuorodomis savo ruožtu gali būti programuojamos išvedimo procedūros. Taip gaunamos “asociatyviosios programos”. Hipertekstą galima sujungti su kitais paieškos metodais, naudojančiais, pavyzdžiui, įprastą informacijos paiešką ir loginius operatorius.

Hiperteksto metodai glaudžiai siejasi su teisinio samprotavimo procesu, kadangi jo esminė dalis yra žinių (sąvokų, normų, bylų, teisinės literatūros ir kt.) susiejimas tarpusavyje. Galima išskirti tokius hiperteksto panaudojimo aspektus kompiuterinėse teisės sistemose, kaip informacijos paieška, žinių fragmentų susiejimas, teisinės medžiagos pateikimas ir valdymas, išvedimų paaiškinimas (pavyzdžiui, sujungiant su logine teisinių samprotavimų sistema), teisinis mokymas [39, p. 298].

Praktinis eksperimentas su hiperteksto panaudojimu intelektualizuotoje teisės sistemoje buvo atliktas Švedijos teisės ir informatikos tiriamajame institute (IRI) pagal VF projektą [39, p. 301–309]. Hipertekstinę sistemą buvo planuota sujungti su taisykline sprendimų palaikymo sistema, kuri buvo kuriama tuo pat metu. Deja, literatūroje nepavyko rasti, ar integruota sistema buvo sukurta iki galo. Kadangi tikslas buvo iširti, kiek ši metodika iš viso tinka teisės srityje, eksperimentui buvo paimtas mažas darbo teisės fragmentas, nagrinėjantis darbuotojo atleidimo dėl pravaikštų aspektą. Sprendimų palaikymo sistemoje daugiausia dėmesio buvo skirta negriežtai sąvokai “pateisinama priežastis”, kurios prasmė nustatoma tik iš ankstesnių bylų ir kitokių dokumentų, kurie nėra teisiniai aktai. Hipertekstinė duomenų bazė buvo pradėta kurti nuo pagrindinių sąvokų apibrėžimų ir žinių bazės struktūros sukū-

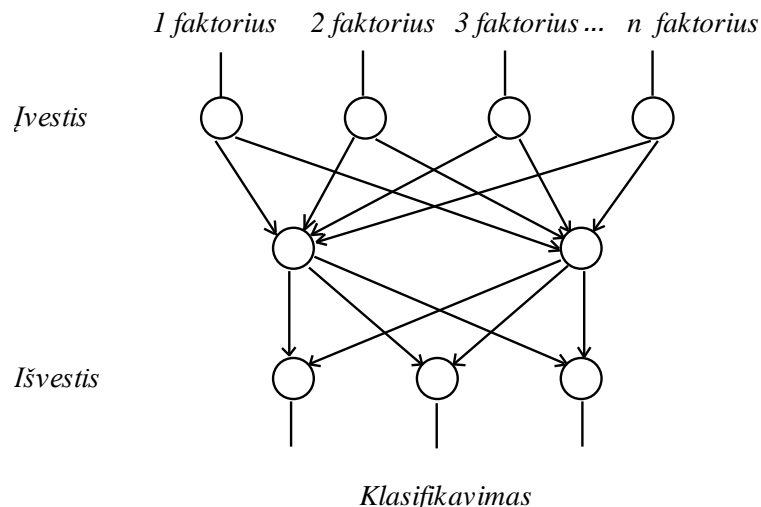
rimo. Plečiant sistemą, buvo įtraukti atitinkamų teisinių aktų tekstai, ankstesnių bylų santraukos ir visi tekstai, turinio ir raktinių žodžių puslapiai, realizuota teksto paieška, pagalba ir paaiškinimai naudotojui, galimybė naudotojui įvesti savo pastabas, numatytas ryšys su teisinės literatūros biblioteka ir kitomis tolimosiomis duomenų bazėmis.

Be šio eksperimento, 1991 m. IRI buvo pradėtas LexLab projektas, pagal kurį kuriama hipertekstinė sistema, kurioje būtų galima užrašyti taisykles (tokias, kokios būdingos žinių bazėms) [39, p. 309]. Darbe [41] aprašomas hiperteksto panaudojimas rengiant teisinius dokumentus. Ten hipertekstas naudojamas skirtingoms dokumento (įstatymo arba precedento) versijoms saugoti, manipuluoti ir susieti fragmentus, kurių pakeitimas gali turėti įtakos vienas kitam.

7. Neuroniniai tinklai

Dirbtiniai *neuroniniai tinklai* yra ypatingas skaitinis (nesimbolinis) analoginio samprotavimo būdas [42]. Tuo jis labai skiriasi nuo iki šiol apžvelgtų simbolinių metodų. Neuroniniais tinklais neįmanoma modeliuoti pačių samprotavimų, nes prielaidas ir išvadas sieja tik skaitiniai ryšiai. Tačiau juos verta apžvelgti, nes tai yra įdomus dirbtinio intelekto metodas, kurį galima taikyti precedentinės teisės sistemose.

Neuroniniuose tinkluose suprastintai imituojama biologinė nervų sistema, taip pat ir žmogaus smegenys. Juose taikomi statistiniai metodai. Neuroniniai tinklai tinka tuo atveju, kai yra sukaupta daug duomenų su rezultatais, tačiau nėra aiškių taisyklių, kaip iš pradinių duomenų gaunamas rezultatas. Todėl jie pasiteisino sprendžiant daugelį uždavinių, kurie pasirodė per sudėtingi atpažinti ekspertines sistemas, ypač vaizdus, kalbos ir rankraštinius tekstus. Visi šie uždaviniai susiję su šablonų atpažinimu ir klasifikavimu.



Neuroninį tinklą sudaro *neuronai* (paprasti procesoriai, kartais dar turintys nedidelę vidinę atmintį) ir *ryšiai* tarp jų, kuriais perduodamos skaitinės reikšmės. Jei reikšmė viršija tam tikrą ribą (*slenkstį*), tai sužadinamas kitas neuronas tinkle ir t.t. Reikšmės priklauso nuo *svorių*, priskirtų ryšiams. Kai neuroninis tinklas apmokamas pateikiant jam testinius duomenis, šie svoriai automatiškai derinami pagal tam tikras formules. Taip tinklas pritaikomas vienos klasės uždaviniams spręsti.

Neuroniniai tinklai gali būti naudojami precedentinėje teisėje, tačiau į juos negalima įtraukti statutinio pagrindo arba teismų logikos elementų. Todėl jie labiau tinka išankstiniams greitiems bylos rezultatams numatyti (pavyzdžiui, advokatu prieš imantis bylos), bet ne rezultatams pagrįsti. Taip pat galima tikrinti variantus "kas jeigu": įtraukiant arba neįtraukiant vieną ar kitą atributą galima nustatyti esminį atributą, nuo kurio priklauso bylos sprendimas.

Neuroninių tinklų pranašumas yra tai, kad jie patys "išmoksta" įvertinti skirtingų faktorių svarbą. Tuo tarpu įprastame precedentiniame samprotavime tai turi nustatyti ekspertas. Pagrindinis neuroninių tinklų trūkumas yra tai, kad jie negali pagrįsti arba paaiškinti rezultatų.

D. Hunterio nuomone [42], jie nėra paplitę taikant teisę ir dėl nepakankamo pačios technologijos suvokimo, ir dėl abejonių, ar pastangos bus vertos rezultato. Sistemų, naudojančių neuroninius tinklus, pavyzdžiai – PROLEX, kur pasitelkus neuroninius tinklus sprendžiamos negriežtų sąvokų problemos, NEUROLEX, skirta municipalinei teisei (prognozuoja Prancūzijos valstybės tarybos sprendimą dėl vietinės valdžios potvarkių), SCALIR, skirta tinkamiems precedentams išrinkti.

8. Kompiuterinis modeliavimas rengiant teisinius dokumentus

Rengiant teisinius dokumentus pagrindiniai dirbtinio intelekto metodų taikymo tikslai yra palengvinti šių dokumentų rengėjų darbą ir pagerinti rengiamų dokumentų kokybę. Kokybė čia suprantama visų pirma kaip suderinamumas ir išsamumas.

Be bendrų teisinių žinių ypatumų, rengiant teisinius dokumentus atsiranda papildomų ypatumų:

1. Gali būti kelios rengiamo dokumento versijos, todėl žinių bazė gali būti sudaryta iš atskirų fragmentų.
2. Užklausų spektras yra labai platus – tai ir užklausos apie testinės situacijos teises pasekmes, ir metalygio užklausos, ir užklausos apie suderinamumą ir išsamumą.
3. Kitaip suprantama semantinio neapibrėžtumo problema – klausimas yra kaip sumažinti tokio neapibrėžtumo laipsnį, o ne tai, ar galima konkrečią situaciją kvalifikuoti pagal tam tikrą negriežtai apibrėžtą sąvoką (ši klausimą galima palikti naudotojui, reikalaujant užklausoje vartoti tik įstatymo sąvokas).

Galima išskirti skirtingus kompiuterinio modeliavimo, naudojamo rengiant teisinius dokumentus, aspektus:

- automatizuotas ir pusiau automatizuotas transformavimas iš natūralios kalbos į formalizuotą ir atgal,
- formalizuota forma užrašyto dokumento suderinamumo ir išsamumo tikrinimas,
- dokumento gyvavimo ciklo ir grupinio darbo palaikymas,
- socialinių ir kitokių pasekmių įvertinimas taikant imitavimo metodus,
- lyginamoji analizė – skirtingų dokumento variantų palyginimas tarpusavyje arba su kitų šalių analogiškais dokumentais.

Toliau apžvelgsime, kaip šie aspektai yra realizuojami skirtingose sistemose.

Kaip jau minėjome, dar 1957 m. teisininkas L. Allenas [10, p. 833–879] pasiūlė vartoti formalizuotą logikos kalbą rengiant ir analizuojant teisinius dokumentus bei mechaniškai išvedant pasekmes. Vėliau jis įvedė teisinių dokumentų *normalizuotą formą*, kuri skirta sintaksiniam dviprasmiškumams surasti ir pašalinti. Pavyzdžiui, “A ir B arba C” galima suprasti kaip “(A ir B) arba C” arba kaip “A ir (B arba C)”. L. Alleno normalizuotoje formoje naudojamos loginės jungtys (AND, OR, NOT, IF, IFF) ir kai kurie dokumento struktūrizavimo susitarimai. Buvo sukurtos kompiuterinės programos teisiniams dokumentams normalizuoti.

L. Alleno pasiūlymais grindžiama ir sistema Prodeon, kurią minėjome skyriuje apie dedukciją. Autoriai pasiūlė taikyti sistemą jau parašytiems įstatymų projektams tikrinti, formalizuojant juos pusiau automatiškai. Formalizuojama kas žingsnis, pakeliui ieškoma kai kurių nesuderinamumų. Vėliau jie ieškomi analizuojant naudotojo pateiktų testų išvadas. Be to, tikrinama, ar yra numatyti sprendimai situacijoms, kai pažeistos deontinės normos.

P. Mariani ir D. Tiscornia modelyje [43] iš pradžių naudojama tekstų redagavimo programa dokumento tekstui įvesti ir gramatikai patikrinti. Po to ekspertinė sistema (ESPLEX) tikrina normalizuotos dokumento formos semantinį suderinamumą su kitomis teisinėmis normomis tiek dokumento viduje, tiek išorėje. Taip pat galima patikrinti pasekmes taikant įstatymą testinėms situacijoms. Panašiais į P. Mariani ir D. Tiscornia modelio principais sukurtos ir kai kurios kitos sistemos – LEDA, ExpertiSZe, TRACS.

Sistemoje Prologa formalizuota įstatymų užrašymo forma – sprendimų lentelės [44, p. 71–80]. Todėl ši sistema siūloma tik toms naujai kuriamoms teisinėms normoms, kurios yra

sprendimo taisyklės, t.y. išvados jose priklauso nuo kelių sąlygų kombinacijos. Taikant įprastus sprendimų lentelių tikrinimo metodus, tikrinamas jų išsamumas ir suderinamumas. Tiriamos galimybės automatiškai transformuoti sprendimo lenteles į natūralios kalbos sakinius, pasitelkus logikos ir lingvistikos metodus.

Darbe [45, p. 11–22] siūloma formalizavimo kalba – normų freimai, kurie turėtų metodiškai padėti geriau suformuluoti įstatymą. Normų freime reikalaujama užpildyti standartinius slotus (straipsnio tikslas, objektas ir pan.), kurių reikšmės ne visada tiesiogiai nurodomos straipsnyje, nes dalis informacijos arba aprašoma kitur, arba aiški iš konteksto, arba laikoma savaime suprantama. Problemos, atsirandančios pildant šiuos slotus, rodo, kad įstatymas nėra pakankamai aiškus.

Teisinio dokumento rengimo ciklas yra panašus į programinės įrangos kūrimo ciklą, todėl pagal analogiją įvestas terminas “teisės inžinerija”. Pagrindinės teisinio dokumento rengimo stadijos yra analizė, projektavimas, konstravimas (teksto rengimas, svarstymas ir priėmimas atitinkamose institucijose) ir diegimas. Šiam procesui automatizuoti siūloma naudoti standartines projekto valdymo sistemas, priderinant jas prie taikymo srities. Daugumoje tyrimų nagrinėjama teisinio dokumento konstravimo stadija, nors T. J. M. Bench-Caponas dar 1987 m. aprašė projektą [46, p. 181–189], kurio tikslas buvo padėti priešprojektinėje teisinio dokumento rengimo stadijoje, kai formuluojami jo tikslai ir strategijos.

Aktuali problema yra socialinių ir kitokių pasekmių, kurios gali kilti priėmus įstatymą, įvertinimas. Pavyzdžiui, darbe [47, p. 53–60] aprašytas mikroimitacijos taikymas socialinio draudimo įstatymų srityje. Įstatymas normalizuotu pavidalu saugomas žinių bazėje, ir jo veikimas imituojamas, taikant jį statistiškai parinktam situacijų rinkiniui. Rezultatas statistiškai apibendrinamas visiems gyventojams.

Rengiant įstatymus, dažnai reikalingas skirtingų jų variantų palyginimas bei palyginimas su kitų šalių analogiškais dokumentais. Tai sunkus uždavinys, ypač kai įstatymas yra didelės apimties ir sudėtingos struktūros. Intelektualizuotos sistemos siūlo normalizuotu pavidalu žinių bazėse saugomų įstatymų palyginimą, naudojant testines situacijas. Darbe [47, p. 53–60] naudojama mikroimitacija, kuri leidžia palyginti naują įstatymą su ankstesniu, įvertinant jų pasekmes. Darbe [48, p. 107–114] taip pat lyginamos testinės situacijos, tačiau čia lyginimo tikslas yra nustatyti nesuderinamumą, ir siūloma taikyti metataisyklių lygį imituojant teisininkų mąstymą, kai reikia pasirinkti vieną iš kelių nesuderinamų išvadų.

Išvados

Apžvelgus dirbtinio intelekto metodus, taikomus modeliuojant teisinius samprotavimus, galima padaryti tokias išvadas:

1. Pagrindiniai samprotavimų modeliavimo metodai yra dedukcija ir precedentinis samprotavimas, taikant indukcijos ir analogijos metodus. Šie metodai nėra alternatyvūs, greičiau jie papildo vienas kitą. Sistemose, modeliuojančiose teisinius samprotavimus, dažniausiai jie yra derinami vienas su kitu ir taip gaunami hibridiniai metodai. Vienas iš pagrindinių tokių sistemų skirtumų yra tai, koku mastu jose taikomas vienas ar kitas samprotavimų modeliavimo metodas ir koku būdu jie jungiami vienas su kitu. Dažniausiai dedukcija taikoma teisinėms normoms ir euristikoms modeliuoti. Tai yra pagrindinis metodas statutinės teisės šalyse, prie kurių priskiriama ir Lietuva. Indukcija taikoma konkreitiems atvejams apibendrinti ir tokiu būdu gauti dedukcijos taisyklės. Analogija taikoma tada, kai norima priimti sprendimus, atsižvelgiant į buvusius precedentes (sprendimus, priimtus panašiais atvejais).
2. Bendri dirbtinio intelekto ir teisės tyrimai yra naudingi abiem pusėms. Viena vertus, kuriamos naudingos teisininkams sistemos ir formalios metodikos. Kita vertus, teisė yra ideali analizės, tikrinimo ir eksperimentų sritis dirbtinio intelekto tyrimams.
3. Socialiniai teisės aspektai riboja teisinių samprotavimų automatizavimo galimybes, todėl kompiuterinio modeliavimo eksperimentai sulaukia kai kurių teisininkų kritikos. Tačiau pripažįstama, kad tam tikru mastu kompiuterinis teisinių samprotavimų modeliavimas yra įmanomas ir gali padėti teisininkams priimant sprendimus.



LITERATŪRA

1. **Loevinger L.** Jurimetrics: The Next Step Forward. – Minnesota Law Review. 33, 1949.
2. **Mehl L.** Automation in the Legal World: From the Machine Processing of Legal Information to the 'Law Machine' // "Mechanisation of Thought Processes", HMSO: – London, 1958.
3. **Sartor G.** Artificial Intelligence and Law. Legal Philosophy and Legal Theory. – CompLex 1/93, Tano, Oslo, 1993.
4. **Čaplinskas A., Paliulionienė L., Sinkevičiūtė S.** Project NEMESIS: Knowledge-based aspects in law engineering systems // Norman Revell, A Min Tjoa (eds.), "Systems Applications, 6th International Conference. Workshop Proceedings" (DEXA-95 proceedings). – London, 1995.
5. **MacKenzie D.** The Automation of Proof: A Historical and Sociological Exploration. – IEEE Annals of the History of Computing, 1995. Vol. 17. No. 3.
6. **Robinson J. A.** A Machine-Oriented Logic Based on the Resolution Principle. – Journal of the Association for Computing Machinery. 1965. Vol. 12. No. 1.
7. **Reiter R.** On Closed World Databases // Gallaire H. and Minker J. (eds.) "Logic and Databases". – New York: Plenum Press, 1978.
8. **McCarty L. T., Sridharan N. S.** The representation of an evolving of legal concepts: I. Logical templates // Proceedings of the 3rd Biennial Conference of the Canadian Society for Computational Studies of Intelligence (Victoria B. C.), Canadian Society for Computational Studies of Intelligence, 1980.
9. **Minker J.** An Overview of Nonmonotonic Reasoning and Logic Programming. The Journal of Logic Programming. 1993. Vol. 17.
10. **Allen L. E.** Symbolic logic: a razor edged tool for drafting and interpreting legal documents. – Yale Law Journal 66, May 1957.
11. **McCarty L. T.** Reflections on TAXMAN: An Experiment in Artificial Intelligence and Legal Reasoning. – Harvard Law Review. 1977. Vol. 90.
12. **Sergot M. J., Sadri F., Kowalski R. A., Kriwaczek K., Hammond P., Cory H. T.** The British Nationality Act as a Logic Program. – Communications of the ACM. 1986. Vol. 29. No. 5.

13. **Jones S., Mason P., Stamper R.** LEGOL 2.0: A relational specification language for complex rules. – Information Systems. 1979. Vol. 4. No. 4.
14. **Gordon T. F.** OBLOG-2: A Hybrid Knowledge Representation System for Defeasible Reasoning. // “The First International Conference on Artificial Intelligence and Law: Proceedings of the Conference”. – New York: ACM Press, 1987.
15. **Greinke A.** Legal Expert Systems: A Humanistic Critique of Mechanical Legal Inference. – Honours, Australian National University, 1992. (http://cleo.murdoch.edu.au/asu/edtech/cleo_web/mdomo/elaw-j.arcs/greinke.txt).
16. **Bench-Capon T. J. M., Sergot M. J.** Toward a Rule-Based Representation of Open Texture in Law. Chapter Six // C. Walter (ed.) “Computer Power and Legal Language”, Quorum: – London, 1988.
17. **Groendijk C., Herrestad H.** An Incremental Approach to Legal Drafting Support // J. S. Svensson, J. G. J. Wassink, B. van Buggenhout (eds.) “Legal Knowledge Based Systems: Jurix'93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law”. – Lelystad: Koninklijke Vermande, 1993.
18. **Barklund J., Hamfelt A.** Hierarchical representation of legal knowledge with metaprogramming in logic. – The Journal of logic Programming. 1994. Vol. 18.
19. **Waterman D. A., Peterson M. A.** Rule-based Models of Legal Expertise. – Proceedings of AAAI-80. – Stanford, CA, 1980.
20. **Greenleaf G.** Australian Approaches to Computerising Law – Innovation and Integration. – Australian Law Journal. 1991. Vol. 65.
21. **Michaelsen R. H.** An Expert System for Federal Tax Planning. – Expert Systems. 1984. Vol. 1. No. 2.
22. **Smith J. C., Deedman C.** The Application of Expert Systems Technology to Case-Based Law. – Proceedings of the First International Conference on Artificial Intelligence and law. – New York, ACM Press, 1987.
23. **Bench-Capon T. J. M.** Deep Models, Normative Reasoning and Legal Expert Systems. – Proceedings of the Second International Conference on Artificial Intelligence and Law, ACM Press, 1989.
24. **Ashley K. D.** Modeling Legal Argument: Reasoning with Cases and Hypotheticals. – Cambridge, MA: MIT Press, 1990. (Revision of Ph. D. thesis)
25. **Popple J.** A pragmatic legal expert system. – Applied Legal Philosophy Series. – Dartmouth (Ashgate), Aldershot, 1996 (<http://cs.anu.edu.au/software/shyster/book/>).
26. **Rissland E. L., Skalak D. B.** CABARET: Rule Interpretation in a Hybrid Architecture. – International Journal of Man-Machine Studies, 1991. Vol. 34. No. 6.
27. **Prakken H., Sartor G.** Argument-based logic programming with defeasible priorities. – Journal of Applied Non-classical Logics. 1997. Vol. 7. No. 1.
28. **Loui R. P., Norman J., Olson J., Merrill A.** A design for reasoning with policies, precedent, and rationales. – Proceedings of Fourth International Conference on Artificial Intelligence and Law. – Amsterdam: ACM Press, 1993.
29. **Gordon T. F.** The Pleadings Game. An Artificial Intelligence Model of Procedural Justice. – Kluwer, 1995.
30. **Lodder A. R., Herczog A.** DiaLaw: A dialogical framework for modelling legal reasoning. – Proceedings of the Fifth International Conference on Artificial Intelligence and Law, College Park, MD: ACM Press.
31. **Berman D. H., Hafner C. D.** Representing Teleological Structure in Case-Based Legal Reasoning: The Missing Link. – Proceedings of the Fourth International Conference on Artificial Intelligence and Law: ACM Press, 1993. (http://www.slw.neu.edu/public/home/clinics/ai_law/teleo6ww.htm)
32. **Prakken H., Sartor G.** A dialectical model of assessing conflicting arguments in legal reasoning. – Artificial Intelligence and Law. 1996. Vol. 4. No. 3/4.
33. **Bench-Capon T. J. M.** Argument in Artificial Intelligence and Law // J. C. Hage, T. J. M. Bench-Capon, M. J. Cohen, H. J. van den Herik (eds.) “Legal Knowledge Based Systems: Jurix'95: Telecommunication and AI & Law”. – Lelystad: Koninklijke Vermande, 1995. (http://130.89.90.84/papers_html/j95-html/j95-1.html)
34. **Wieringa R. J., Meyer J.-J. Ch.** Applications of Deontic Logic in Computer Science: A Concise Overview // J. J. C. Meyer, R. J. Wieringa (eds.) “Deontic Logic in Computer Science: Normative System Specification”. – New York, Wiley, 1993.

35. **Smith T.** Legal Expert Systems: Discussion of Theoretical Assumptions. – CompLex 3/95, Tano. – Oslo, 1995.
36. **Biagioli C., Mariani P., Tiscornia D.** ESPLEX: A Rule and Conceptual Model for Representing Statutes // “The First International Conference on Artificial Intelligence and Law: Proceedings of the Conference”. – New York: ACM Press, 1987.
37. **Allen L. E., Saxon C. S.** Analysis of the logical structure of legal rules by a modernized and formalized version of Hohfeld fundamental legal conceptions // A. A. Martino, F. S. Natali (eds.) “Automated Analysis of Legal Texts”, North-Holland, 1986. Edited versions of selected papers from the Second International Conference on Logic, Informatics, Law. – Florence, Italy, September 1985.
38. **Groendijk C., Herrestad H.** An Incremental Approach to Legal Drafting Support // J. S. Svensson, J. G. J. Wassink, B. van Buggenhout (eds.) “Legal Knowledge Based Systems: Jurix'93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law”. – Lelystad: Koninklijke Vermande, 1993.
39. **Wahlgren P.** Automation of Legal Reasoning. A Study on Artificial Intelligence and Law. Kluwer Law and Taxation Publishers. – Deventer, 1992.
40. **Herrestad H.** Is modal logic necessary or to be desired for expert systems in law? – Presented at June 14, 1996, at the occasion of Herrestad's doctoral disputation at the Faculty of Law, at the University of Oslo. (http://www.jus.uio.no/iri/afin/dok/pub/artikler/2_96_b.html).
41. **Leung R.** Versioning on legal applications using hypertext. – Proceedings of the Workshop on Versioning in Hypertext Systems, ACM European Conference on Hypermedia Technology (ECHT'94). – Edinburgh, September 18-23, 1994.
42. **Hunter D.** Commercialising legal neural networks. – The Journal of Information, Law and Technology. 1996. No. 2. (<http://elj.warwick.ac.uk/elj/jilt/artifint/2hunter/>).
43. **Mariani P., Tiscornia D.** Expert Systems as an Aid to Legislative Drafting. – IDG of CNR (Istituto per la documentazione giuridica del Consiglio Nazionale delle Ricerche – Institute for Legal Documentation of the Italian National Research Council), 1990. (Mariani P., Tiscornia D. “I sistemi esperti di supporto alla redazione di testi legislativi” // Atti del IV Congresso internazionale “Informatica e regolamentazioni giuridiche”. – Roma, 1988, sess. X, n. 9.
44. **Van Buggenhout T., Vanthienen J., Schepers J., van Buggenhout B., Wets G., de Smedt L.** The Decision Table Technique as part of a Computer Supported procedure of Legal Drafting // J. S. Svensson, J. G. J. Wassink, B. van Buggenhout (eds.) “Legal Knowledge Based Systems: Jurix'93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law”. – Lelystad: Koninklijke Vermande, 1993.
45. **Van Kralingen R., Oskamp E., Reurings E.** Norm Frames in the Representation of Laws // J. S. Svensson, J. G. J. Wassink, B. van Buggenhout (eds.) “Legal Knowledge Based Systems: Jurix'93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law”. – Lelystad: Koninklijke Vermande, 1993.
46. **Bench-Capon T. J. M.** Support for policy makers: Formulating legislation with the aid of logical models. – Proc. First International Conference on Artificial Intelligence and Law. – Boston, May 1987 (ACM Press).
47. **Svensson J. S.** Using Knowledge Based Micro Simulation in Analysing the Application Of Legislation // J. S. Svensson, J. G. J. Wassink, B. van Buggenhout (eds.) “Legal Knowledge Based Systems: Jurix'93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law”. – Lelystad: Koninklijke Vermande, 1993.
48. **Guidotti P., Turchi F.** A Frame-Work for Law Comparison // In: J. S. Svensson, J. G. J. Wassink, B. van Buggenhout (eds.) “Legal Knowledge Based Systems: Jurix'93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law”. – Lelystad: Koninklijke Vermande, 1993.



Computer Modeling of Legal Reasoning

L. Paliulionienė

Institute of Mathematics and Informatics

SUMMARY

The article presents an overview of the history and present state in artificial intelligence and law investigations. It discusses what features of legal knowledge determine specific features of computer knowledge-based legal systems and what problems these systems work on. Computer modelling methods and their use for the modelling of legal reasoning are considered as well. Knowledge-based legal systems usually use hybrid methods that integrate several methods. The main methods are deduction, induction and analogy. Deduction is usually used to model legal norms and heuristics. It is the main method in the countries with statute law traditions, and Lithuania is one of such countries. Induction is used to generalize concrete cases and obtain deductive rules. Analogy is used to make a decision on the ground of previous decisions made in similar cases. Induction and analogy are applied in case-based reasoning. The article also overviews some other methods used to model legal reasoning: argumentation, deontic logic, hypertext, neural networks. Issues of computer modelling in legal drafting are discussed separately.

