

## **VI dalis. Modeliai**

Šioje dalyje aprašytos svarbiausių knygoje naudojamų tipų diagramos. Diagrama arba schema – tai viena iš sistemų modelių rūšių. Čia aprašomi ne tik schemų sutartiniai ženklai, bet ir sistemų analizės taikant aprašytų tipų modelius pagrindai.

## 25. Sistemų dinamikos modeliai

Šiame skyriuje susipažinsime su sistemų dinamika – analizės metodu, kurį taikant aprašomos ir tiriama kintančios, arba kitaip vadinamos dinaminės, sistemos. Išmoksime skaityti sistemų dinamikos diagramas, pateikiamas kituose skyriuose. Išmoksime kurti ir analizuoti sistemų dinamikos modelius, naudotis jais.

### 25.1. Sistema ir modelis

Bet kokia sistema yra tarpusavyje sąveikaujančių elementų visuma. Elementai, susijungdami į sistemą, įgyja naujų bruožų, būdingų sistemai, bet nebūdingų atskiriems jos elementams. Priklausomybė tarp sistemos savybių ir jos elementų savybių dažnai būna sunkiai suvokiama. Pavyzdžiui, jūros bangavimas nėra akivaizdi vandens ir oro molekulių fizikinių ir cheminių savybių pasekmė. Analizuojant sistemos funkcionavimą svarbiau yra aprašyti ryšius tarp elementų, o ne pačius elementus.

Ryšiai tarp elementų sudaro sistemos struktūrą. Dėl ryšių sistema yra daugiau negu ją sudarančių nepriklausomų elementų rinkinys. Kiekvienas sistemos elementas veikia kitus elementus ir yra veikiamas kitų elementų.

Modelis – tai dirbtinė supaprastinta sistema, atkartojanti modeliuojamos sistemos bruožus. Modeliai kuriami tam, kad būtų lengviau nagrinėti realios sistemos bruožus arba elgseną. Sistemos elgsenos imitavimą nagrinėjanti mokslo sritis vadinama sistemų dinamika.

Kiekvienas sistemos elementas gali būti nagrinėjamas kaip atskira sistema. Tokia sistema vadinama didžiosios sistemos posisteme.

Kiekviena sistema turi ribas, atskiriančias ją nuo aplinkos. Sistemos ribos kartais yra akivaizdžiai matomos. Pavyzdžiui, sistema – automobilis. Ribos gali būti menamos. Pavyzdžiui, taksi parkas, kurio automobiliai yra išsibarstę po visą miestą. Taksi parko sistemos elementus sieja fiziniai ryšiai (radijo ryšio sistema), hierarchiniai ryšiai (pavaldumas), kiti formalūs ryšiai (nuosavybė) ir

bendras tikslas (uždirbti vežant keleivius). Kiekviena dinamiškai aprašoma sistema turi tikslą.

Kai kurie elementai sąveikauja su elementais, esančiais sistemos išorėje. Tokia sąveika vadinama išorine. Kuriant sistemos modelį gali būti aprašomi išoriniai ryšiai, tačiau į modelį neturi būti įtraukiami išoriniai elementai.

Jei nagrinėjama sistema sąveikauja su kitomis sistemomis, kartu su tomis sistemomis sudaro didesnes sistemas. Tokios sistemos vadinamos viršsistemėmis. Sistema yra viršsistemės elementas.

Kuriant modelį svarbu tiksliai apibrėžti modeliuojamą sistemą: ne tik jos ribas, bet ir detalizacijos laipsnį. Modeliuojamos sistemos elementai (posistemės) turi būti panašaus sudėtingumo sistemos. Į modelį nereikia įtraukti išorinių elementų ir ryšių, sudarančių viršsistemę. Modelyje nereikėtų aprašinėti ir sistemos elementų vidinių ryšių, sudarančių posistemę.

Sistema, kuri nesikeičia, yra vadinama statine sistema. Besikeičianti sistema yra dinaminė. Jos kitimas per tam tikrą laiką aprašomas pagal sistemos elgseną. Jeigu sistemos elgsena atitinka aprašytus žinomus bruožus, tai sakome, kad sistema yra tam tikro elgsenos tipo. Skirtingose sistemose pasikartojantys elgsenos tipai vadinami archetipais.

## 25.2. Sistemos parametrai

Dinaminės (kintančios) sistemos būseną aprašoma jos kintančiais parametrais. Sistemą galime analizuoti tik tuo atveju, jei galime palyginti parametrų reikšmes skirtingais laiko momentais. Turime galėti tuos parametrus išmatuoti, tačiau dinaminės sistemos parametrų reikšmės svarbios ne tiek tam, kad galėtume įvertinti sistemos būseną šiuo momentu, o tam, kad galėtume palyginti su kitais laiko momentais. Svarbus yra dinaminės sistemos parametrų kitimas.

Sistemą charakterizuoja kintamieji ir konstantos. Nors šie terminai panašia prasme vartojami ir matematikoje, bet jų apibrėžimai

sistemų dinamikoje skiriasi nuo matematinių. Dydžiai, kintantys laikui bėgant, vadinami kintamaisiais. Dydžiai, kurie per analizuojamą laiko tarpą nekinta arba kinta taip nedaug, kad neveikia kitų dydžių, vadinami konstantomis. Tas pats dydis viename modelyje gali būti kintamasis, o kitame – konstanta.

Pavyzdžiui, turistui užsienio šalis yra statinė sistema. Jam svarbus yra skirtingų bankų valiutų keitimo kursas, kuris greičiausiai stipriai nepasikeis kelionės metu. Nagrinėjant šalies ekonomiką dinamiškai nebesvarbus valiutų kursas, bet labai svarbus to kurso kitimas ir infliacija.

Kai kuriamas sistemų dinamikos modelis skirstant dydžius į kintamuosius ir konstantas įvertinami ir susiję priežastiniai ryšiai. Jei sistemos pokyčiai neveikia dydžio pasikeitimų, jis tokioje sistemoje gali būti interpretuojamas kaip konstanta.

Atvaizduojant modelį kintamuosius ir konstantas svarbu aprašyti taip, kad kitas asmuo galėtų vienareikšmiškai suprasti jų prasmę. Kiekvienas aprašomas keliais žodžiais, nurodančiais parametro esmę, bet ne dabartinę būseną. Būseną nurodo parametro reikšmė. Pavadinimas turėtų tikti bet kokiai parametro reikšmei. Iš pavadinimo turėtų būti aiškūs galimi matavimo vienetai. Jei taip suformuluoti neišeina, tada rekomenduotina matavimo vienetus nurodyti.

Pavyzdžiui:

- *PHP* programuotojo atlyginimas (litais)
- *Apache* serverių paplitimas (serverių dalis)

### 25.3. Ko saugotis kuriant modelius

Reikėtų vengti apibrėžimų naudojant neiginį, nes tokį modelį būtų sunkiau analizuoti. Pavyzdžiui, sunku greitai atsakyti į klausimą „kaip resursų trūkumo sumažėjimas paveiks specialistų trūkumą?“

Modelis neturi emocinio atspalvio. Modelis – tai matematiškai aprašyta sistema. Skirtingi asmenys turi skirtingų tikslų ir skirtingą

gėrio ir blogio supratimą. Kuriant modelį aprašoma sistema, o ne tikslai. Modelis nesikeičia priklausomai nuo modeliuotojo supratimo, kas yra gerai, o kas blogai.

#### 25.4. Priežastis ir pasekmė

Analizuodami reiškinius paprastai nustatome jų priežastis. Ištyrę reiškinį dažnai kalbame apie jo pasekmes. Priežasties ir pasekmės ryšys gali būti aprašomas įvairiomis kalbos konstrukcijomis, pavyzdžiui, teiginiu „jei A, tai B“.

Sudėtingose sistemose pasitaiko atvejų, kai yra akivaizdu, kad keli veiksniai yra susiję, tačiau sunku nustatyti, kur yra priežastis, o kur pasekmė. Tada tiriama reiškinių eilės tvarka, jų sąveikos mechanizmai.

Būtina atidžiai interpretuoti sąvokas, nes ypač šnekamojoje, bet kartais ir dalykinėje, kalboje padaromos klaidos, kurių neatidus skaitytojas ar klausytojas gali ir nepastebėti, tačiau pažodžiui interpretuojant tokį tekstą būtų sukurtas klaidingas modelis.

Reiškiniai vyksta ne tuo pat metu, bet vienas paskui kitą. Tam, kad įvyktų pasekmės reiškinys, prieš tai turi įvykti priežasties reiškinys. Laiko tarpas tarp priežasties ir pasekmės vadinamas vėlavimu. Priklausomai nuo reiškinių inertiškumo vėlavimas gali būti ir sekundės dalys, ir valandos, dienos, mėnesiai ar net metai. Net jei atrodo, kad keli reiškiniai vyksta vienu metu, pasekmė visada eina po priežasties. Pavyzdžiui, šviesa užsidega tuo momentu, kai paspaudžiame jungiklį, tačiau įmanoma įvertinti laiką tarp priežasties (jungiklio paspaudimo) ir pasekmės (užsidegusios šviesos).

Apibrėžiant priežastį ir pasekmę svarbu ne tik įvykių eilės tvarka, bet ir jų priklausomybė. Kaip anekdotas pasakojamas toks pokalbis:

Visą dieną nebuvo šviesos, todėl gali ištirpti šaldytuvus.

Įvykių seka gali ką nors suklaidinti: pradžioje užgeso šviesa, paskui ėmė tirpti šaldytuvo ledas. Žodis „todėl“ turėtų skirti prieža-

stį ir pasekmę. Ir kalbos konstrukcija, ir įvykių seka rodo priešastinį ryšį, tačiau šviesos užgesimas nėra priešastis. Akivaizdu, kad neveikiantis apšvietimas ir atšilęs šaldytuvas neveikia vienas kito, bet turi vieną bendrą priešastį: namuose nėra elektros. Tai nebūtų taip akivaizdu, jei nežinotume sistemos dėsnį.

Svarbu neapsirikti šnekamąją kalbą paverčiant loginėmis konstrukcijomis. Formuluoatė „jei ..., tai ...“ kartais būna apgaulinga. Ji gali reikšti „jei priešastis, tai pasekmė“, bet gali būti ir „jei matau pasekmę, tai galiu spręsti apie priešastį“.

Pavyzdžiui, panašus loginis teiginys gali būti aprašomas sukeičiant pasekmę ir priešastį vietomis:

Jei yra šalta, tai žmonės šilčiau rengiasi.

arba sukeičiant priešastį ir pasekmę:

Jei žmonės šilčiau rengiasi, tai (matyt) yra šalta.

Pasitaiko žmonių, kurie klaidingai nurodo priešastinį ryšį tikėdamiesi, kad taip jų kalba esą atrodys rimtesnė, moksliškesnė. Pavyzdžiui, pasakoma:

Jei visi kaimynai prie savo namų pasėjo žolę, tai Vytautas prie savo namo pasodino rožes.

Akivaizdu, kad priešastinio ryšio čia nėra. Sakinys yra netaisyklingas ir kalbos požiūriu. Kalboje reikėtų vengti tokių klaidų, o kuriant modelius svarbu neapsirikti interpretuojant tekstus.

Sutartinė ženklų kalba padeda ne tik greičiau užrašyti ir perskaityti tokius teiginius, bet ir išvengti minėto šnekamosios kalbos daugiareikšmiškumo. Priežastinis ryšys žymimas rodykle nuo priešasties link pasekmės.

Ugnis → Dūmai

Nors galime pasakyti

Nėra dūmų be ugnies;

ir

Nėra ugnies be dūmų.

### 25.5. Loginiai priešastiniai ryšiai

Jei tarp dviejų reiškinių yra grynai loginis ryšys, tuomet priešastis gali vykti kartu su pasekme, bet taip pat galima teigti, kad tai nėra priešastis ir pasekmė, o tik du būdai aprašyti tą patį reiškinį.

Pavyzdžiui,

Jei Birutės dukrai gimė vaikas, tai Birutė tapo močiute.

Čia vienas ir tas pats įvykis vienam yra vaiko gimimas, kitam – vaikaičio.

Panašus atvejis yra loginė išvada. Pavyzdžiui,

Jei vidutinis atlyginimas didesnis už du tūkstančius eurų, tai vidutinis atlyginimas didesnis už tūkstantį eurų.

Loginiais ryšiais susieti veiksniai įtraukiami į modelį tik tuo atveju, jei abu veiksniai svarbūs sistemos priešastinių ryšių grandinėms ir sistemos dinamikai.

Sistemos veiksniai yra apibendrinti sistemos parametrai ir netgi esant loginei priklausomybei tokia priklausomybė gali būti nevienareikšmiška.

Pavyzdžiui,

Kuo daugiau gimsta vaikų, tuo daugiau atsiranda senelių.

Toks teiginys gali būti ir neteisingas, jei daugiau vaikų gimsta tik šeimose, jau turinčiose vaikų. Todėl „senelių skaičius“ yra

atskiras veiksnys, logiškai veikiamas tokių veiksmų, kaip „gimstamumas“ ir „vidutinis vaikų skaičius šeimoje“.

### 25.6. Sistemų dinamikos diagrama

Analizuodamas pasaulį žmogus mintyse sukuria jį supančių sistemų modelius. Psichologų teigimu, žmogus negali mintyse aprėpti ilgų priežastinių ryšių grandinių. Vienu metu galvojama tik apie sistemos fragmentą. Nubraižytą schemą galima analizuoti dalimis. Be to, schemą patogiu pateikti kitiems siekiant dirbti kartu ir išsaugoti rezultata. Tiek kokybinio modelio privalumų. Juo remiantis galima sukurti kiekybinį modelį ir kompiuteriu atlikti sudėtingus skaičiavimus prognozuojant sistemos elgseną ir parametrus įvairiose situacijose.

Čia aprašomas modelis susideda iš dviejų tipų elementų. Tai veiksnys ir priežastinis ryšys. Veiksnys yra sistemą charakterizuojantis kintamasis dydis. Priežastinis ryšys nurodo, kaip vieno veiksnio reikšmės pokytis veikia kito veiksnio pokytį, ir žymimas rodykle nuo priežasties link pasekmės. Šalia priežastinio ryšio rodyklės nurodomas poveikio ženklas. Jei priežastinio veiksnio pasikeitimas sukelia tos pačios krypties pasekmės veiksnio pasikeitimą, sakome, kad tai teigiamo pobūdžio priežastinis ryšys, ir žymimas pliusas. Jei pasekmės ženklas priešingas, tada tai neigiamo pobūdžio priežastinis ryšys ir žymimas minusas.

Tokiais paprastais elementais galima aprašyti sudėtingiausias sistemas. Veiksniai išdėstomi lape ir sujungiami priežastinių ryšių rodyklėmis.

### 25.7. Priežastinių ryšių grandinės

Kiekvienas sistemos veiksnys sąveikauja su kitais veiksniais. Veiksnio pokytis yra kitų veiksmų pokyčių pasekmė ir priežastis. Taip veiksniai sudaro priežastinių ryšių grandines.



Investicijos į interneto prieigą → interneto skverbtis → el. prekybos pirkėjų skaičius

Jei tokia grandinė susijungia į žiedą, ji vadinama grįžtamojo ryšio kilpa.

### 25.8. Grįžtamojo ryšio kilpos

Paveikus kilpoje esantį veiksnį, poveikis apeina kilpą ir sugrįžta. Priklausomai nuo grįžtamojo ryšio ženklų kilpa vadinama teigiama arba neigiama.

Analizuojant modelį svarbu įvertinti kurios kilpos veikia. Gali būti, kad kilpa nubraižyta teisingai, tačiau poveikis jos nepraeina, nes pakeliui yra slopinamas stipriau veikiančių kilpų. Kilpos konkuruoja tarpusavyje. Kuo ilgesnė kilpa, tuo didesnė tikimybė, kad poveikis jos neapeis. Kiekviename žingsnyje poveikis vėluoja, todėl svarbu įvertinti, kiek laiko vėluos grįžtamasis ryšys.

### 25.9. Sistemos savireguliacija

Jei grįžtamojo poveikio ženklas pasikeičia, tuomet kilpa vadinama neigiama kilpa. Tokia kilpa kompensuoja poveikį ir sugrąžina sistemą į pusiausvyros būseną (25.1 pav.).



25.1 pav. Stabili sistema

### 25.10. Nestabili sistemos būseną

Jei grįžtamojo poveikio ženklas nepasikeičia, tuomet kilpa vadinama teigiama kilpa. Tai yra nestabilios būsenos kilpa, lemianti eksponentinį parametro didėjimą (25.2 pav.).



25.2 pav. Nestabili sistema

Realiame pasaulyje kiekviena sistema anksčiau ar vėliau įgyja pusiausvyrą, tai yra ima dominuoti savireguliaciją užtikrinančios neigiamos kilpos. Rutuliukas anksčiau ar vėliau pasiekia duobės dugną.

### 25.11. Modelio taikymas

Modelis padeda prognozuoti sistemos elgseną. Jei modelis pasitelkiamas problemai spręsti, lengva atskirti sisteminių sprendimą nuo simptominio, padedančio kompensuoti matomas pasekmes, bet nepadedančio spręsti problemos esmės.

Kilpos padeda arba trukdo sprendimui. Teigiama kilpa stiprina poveikį, todėl gali padėti pasiekti rezultatą mažesnėmis sąnaudomis. Neigiama kilpa gali sumažinti arba visai panaikinti poveikį. Tačiau ir neigiama kilpa gali padėti efektyviai spręsti problemą. Sudarius tinkamas sąlygas nusistovės pageidaujama pusiausvyra.

### 25.12. Literatūra

- Norvaišas, S.; Sruogis, V. *Sistemų dinamikos pradmenys*. Vilnius: Saulius Norvaišas, 2003.  
<http://www.culture.lt/science/SD/sd.htm>

### 25.13. Klausimai ir uždaviniai

1. Remdamiesi žiniasklaida sudarykite aktualių problemų ir siūlomų jų sprendimų sąrašą. Nurodykite, kurie iš sprendimų yra simptominiai.
2. Sukurkite modelį, kuris padėtų spręsti pasirinktą problemą. Pateikite nurodytos struktūros darbo ataskaitą.

Ivade aprašykite modeliuojamą sistemą, problemą ir tikslą.

Problema turi būti aprašoma kaip mūsų netenkinanti veiksnio reikšmė.

Tikslas – probleminio veiksnio reikšmės koregavimas nurodant, per kokį laiką numatoma tą tikslą pasiekti.

Ivado pavyzdys:

*Modeliuojama Lietuvos ekonomika. Problema – per daug vartotojų (90 %) naudojami nelegalia programine įranga. Tikslas – per penkerius metus trečdaliu sumažinti nelegalios programinės įrangos naudojimą.*

Pagrindinė darbo dalis – sistemų dinamikos modelis. Rekomenduojamas veiksmų skaičius nuo 7 iki 10. Prieš diagramą turėtų būti pateikiamas sistemos veiksmų sąrašas nurodant kiekvieno veiksmo matavimo vienetus ir trumpai paaiškinant veiksmus, kurių prasmė nėra akivaizdi iš jų pavadinimų.

Diagrama braižoma naudojant programą *OpenOffice.org Draw*. Priežastiniams ryšiams nurodyti naudojamas ne rodyklės įrankis (*Arrow*), bet jungiamoji linija (*Connector*). Veiksniai turi būti taip išdėstomi, kad priežastinių ryšių linijos nesikirstų.

Darbo išvados pateikiamos rekomendacijos, kaip turėtų būti sprendžiama problema. Rekomendacijos paremtos diagrama. Nurodoma, kokį veiksmį ir kaip reikėtų paveikti siekiant tikslo.

## 26. Duomenų bazės reliacinis modelis

Šis skyrius – tai įvadas į duomenų bazes. Daugelis studentų detaliau nagrinėjo duomenų bazes ankstesniuose kursuose, tačiau tiems, kas kasdieniame darbe nesusiduria su duomenų bazėmis, verta prisiminti. Šios žinios padės suprasti kai kuriuos kitus skyrius ir pravers atliekant kai kurias praktines užduotis.

Duomenų bazė – tai organizuotai sukaupti duomenys, kuriuos galima papildyti, tvarkyti ir atrinkti reikalingus. Paprastai minimos kompiuterinės duomenų bazės, tačiau bibliotekos kartoteka, susidedanti iš stalčiuose surūšiuotų užpildytų kortelių, galėtų būti pavadinta nekompiuterine duomenų baze. Kompiuterinė duomenų bazė susideda iš duomenų ir duomenų bazių valdymo sistemos (DBVS) – programinės įrangos, naudojamos tiems duomenims tvarkyti.

### 26.1. Reliacinė duomenų bazė

Šiuo metu plačiausiai paplitusios reliacinės duomenų bazės. Tai duomenų bazės, susidedančios iš tarpusavyje susietų lentelių. Sukursime pavyzdį – bibliotekos duomenų bazę. Ji susideda iš knygų sąrašo ir skaitytojų sąrašo. Tokie sąrašai vadinami lentelėmis. Kai skaitytojas pasiima knygą, jo skaitytojo bilieta numeris įrašomas prie tos knygos.

Viena lentelės eilutė skirta vienam aprašomam objektui. Ji vadinama įrašu. Lentelės langelis vadinamas lauku. Laukas, viena-reikšmiškai identifikuojantis įrašą (eilutę), vadinamas raktu.

Kiekvienam skaitytojui suteikiamas unikalus numeris, identifikuojantis skaitytojo įrašą. Tai yra skaitytojo lentelės pirminis raktas. Pirminiam raktui keliami šie reikalavimai:

- Privalomumas. Negali būti įrašo su nenustatyta pirminio rakto reikšme.
- Unikalumas. Įrašui suteikiama pirminio rakto reikšmė, kuri nėra, niekada nebuvo ir niekada nebus suteikta jokiam kitam įrašui.

- Nekintamumas. Kartą įrašui suteikta pirminio rakto reikšmė lieka visiems laikams.

Lentelėje „Skaitytojas“ (26.1 lentelė) lauke „Skaitytojo nr.“ nurodomas šiuo metu knygą pasiėmusio skaitytojo numeris. Vienoje lentelėje nurodomas kitos lentelės (26.2 lentelė) pirminis raktas, kaip nuoroda į tos lentelės duomenis.

26.1 lentelė. Duomenų bazės lentelė „Skaitytojas“

Skaitytojo nr.	Vardas	Pavardė	Skaityto knygas
1	Valdas	Adamkus	5
2	Algirdas	Brazauskas	4
3	Jonas	Jonaitis	2, 6
4	Ona	Onaitytė	1, 3

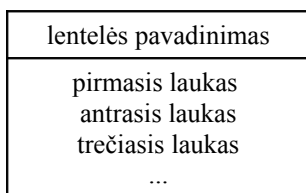
26.2 lentelė. Duomenų bazės lentelė „Knyga“

Knygos nr.	Pavadinimas	Autoriaus vardas	Autoriaus pavardė
1	Ragana ir lietus	Jurga	Ivanauskaitė
2	Vilniaus džiazas	Ričardas	Gavelis
3	Vilniaus pokeris	Ričardas	Gavelis
4	Likimo vardas – Lietuva	Valdas	Adamkus
5	Ir tuomet dirbome Lietuvai	Algirdas	Brazauskas
6	Vienas	Ričardas	Bachas

Šioje supaprastintoje duomenų bazėje nenumatyta, kad galėtų būti keli vienos knygos egzemplioriai.

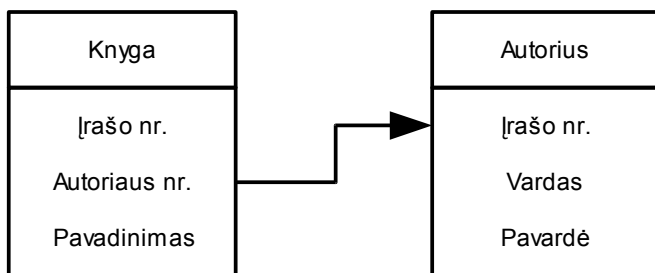
## 26.2. Reliacinis modelis

Kai projektuojamos duomenų bazės, braižomi jų reliaciniai modeliai. Tai schema, kurioje kiekviena lentelė vaizduojama horizontalia linija padalintu stačiakampiu, kurio viršuje nurodomas pavadinimas, žemiau vardinami laukų pavadinimai (26.1 pav.). Ryšiai tarp lentelių vaizduojami rodyklėmis.

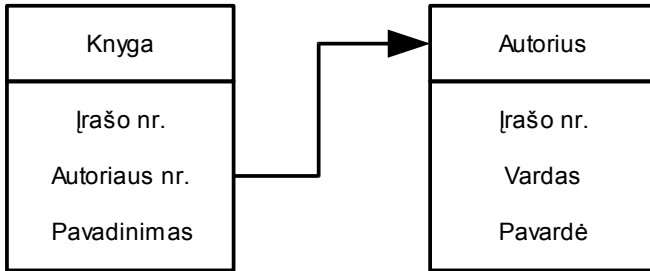


26.1 pav. Lentelės vaizdavimas reliaciniame modelyje

Aprašytus ryšius tarp lentelių nurodo rodyklės. Kaip minėta, identifikatorius yra nuoroda ne į lauką, o į visą įrašą (eilutę, objektą). Todėl kartais žymint tokius ryšius rodyklės brėžiamos ne į raktinį lauką (26.2 pav.), o į lentelės (objekto) pavadinimą (26.3 pav.).



26.2 pav. Duomenų bazės reliacinis modelis  
(rodyklė į pirminio rakto lauką)



26.3 pav. Duomenų bazės reliacinis modelis  
(rodyklė į objektą)

### 26.3. Duomenų bazių kūrimas

Projektuojant duomenų bazes paprastai braižoma reliacinio modelio schema. Ji padeda susikalbėti duomenų bazių specialistams tarpusavyje. Schema reikalinga ir jei duomenų bazė kuriama individualiai. Sukūrus duomenų bazės reliacinį modelį paprastai neskubama kurti duomenų bazės. Modelis padeda optimizuoti duomenų bazės struktūrą. Standartiniai duomenų bazės struktūros optimizavimo veiksmai vadinami duomenų bazės normalizacija.

### 26.4. Duomenų bazių normalizacija

Duomenų bazės normalizacija – tai duomenų bazės pertvarkymas laikantis taisyklių, kurios padeda sumažinti pasikartojimus ir kitus trūkumus.

Normalizacija atliekama pažingsniui derinant duomenų bazę prie aprašytų norminių formų reikalavimų.

### 26.5. Pirmoji norminė forma

**Pirmoji norminė forma (1NF)** reikalauja, kad kiekvienas lentelės laukas turėtų vieną nedalomą reikšmę. Tai reiškia, kad lauko reikšmė negali būti nei aibė, nei sąrašas.

Pirmasis bibliotekos pavyzdys neatitinka 1NF, nes asmens skaitomų knygų laukas yra sąrašas. Duomenų bazė lengvai normalizuojama iš lentelės „Skaitytojas“ pašalinant lauką „Skaito knygas“ ir sukuriant papildomą lentelę „Skaito knygas“, susietą su pirmosiomis dviem lentelėmis (26.3 lentelė).

26.3 lentelė. Duomenų bazės lentelė „Skaito knygas“

Skaitytojo nr.	Knygos nr.	Kada paėmė	Kada grąžino
1	2	2008-01-01	
2	1	2007-12-25	
3	2	2000-09-11	
4	1	2007-09-01	2007-09-02
3	6	2007-09-01	2008-02-14
4	3	2007-09-01	

Normalizuojant lenteles jų struktūra tapo patogi papildomiems svarbiems duomenims – knygų paėmimo ir grąžinimo datoms – įrašyti.

## 26.6. Antroji norminė forma

**Antroji norminė forma (2NF)** reikalauja, kad duomenų bazė atitiktų 1NF reikalavimus ir kad kiekviena lauko reikšmė priklausytų vienam įrašui. Tai yra – pašalinamas duomenų dubliavimas.

Šioje duomenų bazėje kartojasi autorių pavardės. Todėl naudotojui tektų kiekvieną kartą įvedinėti tą patį autorių, o padarius rašybos klaidą sistema nebežinotų, kad tai to paties autoriaus knygos. Todėl normalizuojant duomenų bazę reikėtų sukurti autorių lentelę.



Užuot prie kiekvienos knygos rašę autoriaus vardą, pavardę ir kitus duomenis, mes nurodome tik autoriaus identifikatorių. Rakto reikšmė tarsi įtraukia į knygos lentelę visą autoriaus įrašą (eilutę).

26.4 lentelė. 2NF duomenų bazės lentelė „Knyga“

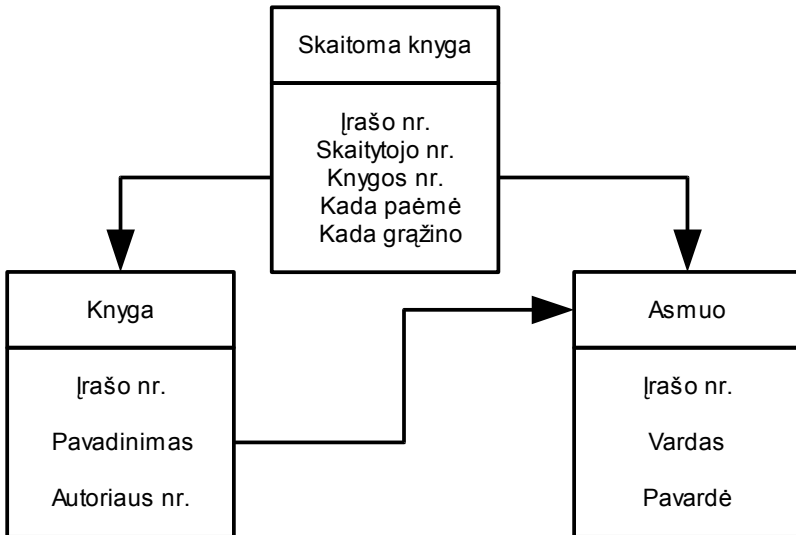
Knygos nr.	Pavadinimas	Autoriaus nr.
1	„Ragana ir lietus“	5
2	„Vilniaus džiazas“	6
3	„Vilniaus pokeris“	6
4	„Likimo vardas – Lietuva“	1
5	„Ir tuomet dirbome Lietuvai“	2
6	„Vienas“	7

26.5 lentelė. 2NF duomenų bazės lentelė „Asmuo“

Asmens nr.	Vardas	Pavardė
1	Valdas	Adamkus
2	Algirdas	Brazauskas
3	Jonas	Jonaitis
4	Ona	Onaitytė
5	Jurga	Ivanauskaitė
6	Ričardas	Gavelis
7	Ričardas	Bachas

Be to, tie patys asmenys įrašyti ir skaitytojų, ir autorių lentelėje. Todėl verta sukurti lentelę „Asmuo“, kurios įrašai būtų nuro-

domi ir „Skaitytojų“, ir „Autorių“ lentelėse (26.4, 26.5 lentelės). Tokios duomenų bazės reliacinis modelis pavaizduotas 26.4 pav.



26.4 pav. Trijų lentelių duomenų bazės reliacinis modelis

### 26.7. Trečioji norminė forma

**Trečioji norminė forma (3NF)** reikalauja, kad duomenų bazė atitiktų 2NF reikalavimus bei kad nė viena neidentifikuojanti lauko reikšmė nebūtų priklausoma nuo kitos neidentifikuojančios lauko reikšmės. Aprašytų ir aukštesnių norminių formų pavyzdžiai detaliau aprašomi duomenų bazių vadovėliuose ir žinynuose.

### 26.8. Literatūra

*O'Reilly* knyga su dviem geniais ant viršelio anglų kalba ir jos vertimas į rusų kalbą yra puikus įvadas į *SQL* duomenų bazes kiekvienam suprantama kalba. Pradedančiajam gali praversti skyriai

apie duomenų bazių projektavimą ir pagrindinių *SQL* komandų aprašai su pavyzdžiais.

Skyrius apie instaliavimą susilaukė kritikos, kai tik knyga buvo išleista, o per dešimtmetį beviltiškai paseno. DBVS apžvalga ir informacija apie kitas programas gali būti įdomi besidomintiems informacinių sistemų istorija:

1. Yarger, R. J.; Reese, G.; King, T.  
*MySQL and mSQL. Databases for Moderate-Sized Organizations and Web Sites.*  
Sebastopol: O'Reilly, 1999.  
ISBN 9781565924345.
2. Яргер, Р.; Риз, Дж.; Кинг, Т. *MySQL и mSQL. Базы данных для небольших предприятий и интернета.*  
Санкт-Петербург: Символ-Плюс, 2000. 560 p.  
ISBN 9785932860106.

Vėlesnėse *O'Reilly* knygose apie duomenų bazes ant viršelio nebeliko vieno iš genijų, o pavadinimuose beliko viena DBVS – *MySQL*. Žinyno *MySQL in a Nutshell* antrajame leidime jau įtraukta tai, ką šioje DBVS pakeitė ją mažiau nei prieš metus įsigijusi korporacija *Sun Microsystems*:

3. Dyer, R. *MySQL in a Nutshell, Second Edition.*  
Sebastopol: O'Reilly, 2008.  
ISBN 9780596514334.

Romo Barono vadovėlis detaliai supažindina skaitytoją su duomenų bazėmis. Jo pirmuosius skyrius galėtų suprasti skaitytojas be išankstinio pasirengimo, o sudėtingesnės temos dėstomos matematinės logikos, aibių teorijų ir programavimo terminais:

4. Baronas, R.

*Duomenų bazių valdymo sistemos.*

*Vadovėlis aukštosioms mokykloms.*

Vilnius: TEV, 2005.

ISBN 9789955680093.

## **26.9. Klausimai ir uždaviniai**

1. Kas yra duomenų bazės lentelė?
2. Kas yra duomenų bazės norminė forma?
3. Ar asmens kodas yra tinkamas asmens įrašo pirminis raktas? Kodėl?
4. Suprojektuokite mokymo įstaigos duomenų bazę, kuri susidėtų iš asmenų lentelės ir su ja susietų dėstytojų lentelės bei studentų lentelės. Nubraižykite lenteles su keliais įrašais kiekvienoje ir duomenų bazės reliacinį modelį.

## 27. Atvirųjų sistemų sujungimo modelis

Sujungus kompiuterius vieni protokolai sudaro sąlygas veikti kitiems protokolams.<sup>65</sup> Tokiai sudėtingai sistemai aprašyti sukurtas *OSI* modelis (angl. *Open Systems Interconnection Reference Model*)<sup>66</sup>. Tai abstraktus ryšio protokolų, naudojamų tinkluose, aprašas.

*OSI* modelis aprašo septynis lygmenis (angl. *layer*) – funkcijų, naudojamų protokoluose, grupes (27.1 pav.). Kiekvienas lygmuo naudojami žemesnio lygmens paslaugomis ir teikia paslaugas aukščiau esančiam lygmeniui. Tradiciškai pirmasis lygmuo yra vaizduojamas apačioje. Todėl ir lygmenų sąrašas dažnai pradedamas nuo septintojo – aukščiausiojo.

Application	Application layer protocol	Application	<b>Upper Layers</b>
Presentation		Presentation	
Session		Session	
Transport	Communication layer protocol	Transport	<b>Middle Layers</b>
Network		Network	
Data Link	Physical topology	Data Link	<b>Lower Layers</b>
Physical		Physical	

27.1 pav. *OSI* lygmenų klasifikacija

Kadangi sąveika vyksta tik tarp gretimų *OSI* lygmenų, tai gerokai supaprastina sudėtingų komunikacijos sistemų kūrimą ir

<sup>65</sup> [http://en.wikibooks.org/wiki/Telecommunications\\_and\\_Network\\_Security](http://en.wikibooks.org/wiki/Telecommunications_and_Network_Security)

<sup>66</sup> [http://en.wikipedia.org/wiki/OSI\\_model](http://en.wikipedia.org/wiki/OSI_model)

aprašymą. Kompiuterių tinklų specialistas privalo labai gerai išmanyti savo lygmenį ir neblogai žinoti, kas vyksta gretimuose lygmenyse. Tolimesniais lygmenimis galima nesidomėti, daugiau laiko skirti savo lygmens problemų ir galimybių analizei.

VII lygmuo – taikomųjų programų lygmuo (angl. *Application layer*) – aukščiausiasis lygmuo, nustato tinklo teikiamas paslaugas vartotojo programoms. Pavyzdžiui: *SSH, telnet, FTP, IRC, echo, VNRP, HTTP, NFS, RTSP, SIP, SMTP, POP3, SNMP, whois, XMPP*.

VI lygmuo – duomenų pateikimo lygmuo (angl. *Presentation layer*) – nustato duomenų kodavimo per sesiją taisykles. Pavyzdžiui: *MIME, XDR, ASN.1, SMB, AFP, NCP*.

V lygmuo – seansų lygmuo (angl. *Session layer*) – aprašo duomenų apsikeitimo tarp galinių sistemų taisykles. Pavyzdžiui: *ASAP, TLS, SSL, ISO 8327 / CCITT X.225, RPC, NetBIOS*.

IV lygmuo – transporto lygmuo (angl. *Transport layer*) – užtikrina skaidrų duomenų perdavimą tarp tinklo vartotojų ir suteikia norimas patikimumo garantijas. Pavyzdžiui: *TCP, UDP, RTP, SCTP, SPX, ATP, IL*.

III lygmuo – tinklo lygmuo (angl. *Network layer*) – aprašo, kaip duomenų sekos turi būti adresuojamos ir perduodamos visame tinkle. Pavyzdžiui: *IP, ICMP, IGMP, IPX, OSPF, RIP, IGRP, EIGRP, ARP, RARP, X.25*.

II lygmuo – ryšio lygmuo (angl. *Data link layer*) – aprašo ryšį tarp gretimų (tiesiogiai sąveikaujančių) tinklo komponentų, Pavyzdžiui, kadrų struktūrą. Pagrindinė funkcija – atskirti bitų sekas, jų pradžią, pabaigą. Pavyzdžiui: *Ethernet, Token ring, HDLC, Frame relay, ISDN, ATM, 802.11 (WiFi), FDDI, PPP, PPPeE*.

I lygmuo – fizinis lygmuo (angl. *Physical layer*) – aprašo fizinius perduodamo signalo ir terpės, kuria jis perduodamas, parametrus (dažnius, moduliacijas, reikalavimus kabeliams, jungtims, duomenų ir fizinio signalo virsmus). Pagrindinė jo funkcija – atskirti 0 ir 1. Pavyzdžiui: *10BASE-T, 100BASE-T, 1000BASE-T, SONET/SDH, G.709, 802.11*, kabeliai ir kita fizinė tinklo įranga.

### 27.1. Kompiuterių sujungimas vietiniame tinkle

**I ir II lygmuo pagal OSI.** Internetas nėra vienas taškas, prie kurio būtų jungiami kompiuteriai. Internetas yra tarpusavyje sujungtų kompiuterių tinklų visuma. Kompiuteris visų pirma jungiamas prie vietinio tinklo (*LAN, local area network*). Būtent tokį prisijungimą užtikrina žemiausi *OSI* hierarchijos lygmenys. Fizinį tinko sujungimą aprašantys standartai leidžia kiekvienam pagaminti įrenginį, kuris galėtų būti prijungtas prie tinklo.

Prisijungimo standartų naudą iliustruoja mobiliųjų telefonų istorija. Kiekvienas *GSM* telefonų gamintojas kūrė kroviklių ir išorinių ausinių jungtis. Vartotojas, įsigijęs naują telefono aparatą, nebegali panaudoti įrangos, kuri buvo pritaikyta kitam telefonui. Vartotojai įvertino standarto naudą, kai pradėtos gaminti ausinės ir telefonai, palaikantys *Bluetooth* standartą.

### 27.2. Kompiuterių sąveika tinkle

*TCP/IP* – populiariausia sudėtinių tinklų organizavimo priemonė. Pagrindinis tinklo lygmens protokolas yra *IP* protokolas (*Internet Protocol*). Jis buvo kurtas paketiniams duomenims perduoti sudėtiniais tinklais, kuriuos sudaro daug vietinių tinklų, susijusių vietiniais ir globaliais ryšiais. Todėl *IP* protokolas gerai veikia sudėtingos topologijos tinkluose, racionaliai išnaudoja jų posistemes ir ekonomiškai išnaudoja lėtų ryšio linijų pralaidumo gebą.

*TCP/IP* naudojami trijų tipų adresai: *MAC* (vietinis, aparatinis), *IP* adresas ir simbolinis vardas.

### 27.3. Kompiuterių fiziniai adresai

**II lygmuo pagal OSI.** Jei tinklas yra nedidelis ir paprastas, tada paprasčiausia yra naudotis kompiuterių *MAC* adresais. Tokį unikalų adresą kiekvienam įrenginiui suteikia gamintojas. Šis adresas nenurodo, kur kompiuteris yra prijungtas. Nedideliame tinkle visi kompiuteriai ar kiti įrenginiai „girdi“ vienas kitą bei pagal nurodytą *MAC* adresą kiekvienas iš jų atsirenka sau skirtus

duomenų paketus. Tai analogiška atvejui, kai kambaryje vienas žmogus vardu kreipiasi į kitą girdint ir aplinkiniams, esantiems kambaryje. Ryšys užtikrinamas tik tada, jei kalbama po vieną. Plečiant tokį tinklą kompiuteriai ima vienas kitam trukdyti. Tai yra viena iš priežasčių, kodėl tinklas turėtų būti skaidomas į potinklius, o paketai perduodami tik tam tinklui, kuriam skirti. Kita priežastis – ryšio kanalų apkrovos valdymas.

#### 27.4. Kiekvienas kompiuteris tinkle turi *IP* adresą

**III lygmuo pagal OSI.** Minėtas problemas sprendžia *IP* (internetu protokolus). Jis kiekvienam įrenginiui suteikia *IP* adresą (numerį), susidedantį iš tinklo adreso ir kompiuterio adreso tame tinkle. *IP* adresą sudaro 4 baitai, paprastai užrašomos jų dešimtainės reikšmės be tarpų, atskirtos taškais. Pavyzdžiui, „192.0.2.119“. Ši adresų sistema vadinama *IPv4*. Toks *IP* adresas yra 32 bitų seka. Konfigūruojant kompiuterius ir kitus įrenginius nustatomi jų *IP* adresai.

Prenkant *IP* adresus turėtų būti atsižvelgiama į kai kuriuos apribojimus. Mazgo adrese neturėtų būti skaičių, kurie dvejetainėje sistemoje atvaizduojami tik nuliais (00000000) arba tik vienetais (11111111). Tai – rezervuotos reikšmės: nulis (00000000) išreiškia esamo tinklo numerį, o 255 (11111111) naudojamas transliavimo pranešimui siųsti, todėl adresuoti praktiškai galima ne 256 mazgus, o  $256 - 2 = 254$ . Tinklo maršrutizatoriui, per kurį tinklo įrenginiai jungiasi prie kitų tinklų, dažnai suteikiamas pirmasis tinklo adresas.

Atskirai nurodoma, kokia *IP* adreso dalis yra tinklo numeris, likusi dalis yra mazgo numeris. Yra trys būdai tai nurodyti. Gali būti nurodoma tinklo klasė, tinklo numeriui skiriamų bitų skaičius arba *IP* adreso kaukė (*netmask*).

*RFC1466* aprašo tinklo skirstymą į A, B ir C klases pagal pirmuosius bitus<sup>67</sup> (27.1 lentelė), o jį pakeitęs *RFC2050* daugiau

<sup>67</sup> <http://tools.ietf.org/html/rfc1466>



dėmesio skiria taupiam *IPv4* adresų paskirstymui, reikalingam dėl jau juntamo adresų trūkumo<sup>68</sup>.

27.1 lentelė. *IP* tinklų tipai

Tinklo klasė	A	B	C
Pirmieji bitai	0...	10...	110...
Pirmasis baitas	0 – 127	128 – 191	192 – 223
Tinklo adreso ilgis	8 bitai	16 bitų	24 bitai
Kaukė	255.0.0.0	255.255.0.0	255.255.255.0
Vieno tinklo dydis	16 777 214	65 534	254

Gali būti ir kitaip paskirstomi *IP* adreso bitai. Pavyzdžiui, tinklas mažesnis už C klasės tinklą, turintis iki 32 unikalių *IP* adresų: tinklo numeris kartu su nurodytu tinklo dydžiu – 192.0.2.0/27, to tinklo kaukė – 255.255.255.224, vienas iš tokio tinklo *IP* adresų – 192.0.2.11, šio tinklo paskutinis adresas 192.0.2.31.

### 27.5. Specialios *IP* adresų grupės

Ne visada kiekvienam kompiuteriui tinkle suteikiamas unikalus visame internete *IP* numeris. *RFC1918* aprašo vietiniams tinklams rezervuotas adresų grupes ir jų naudojimo principus<sup>69</sup> (27.2 lentelė). Įrenginys, turintis tinklo vidinį adresą, gali jungtis prie kitų interneto resursų ir būti prieinamas iš interneto per maršrutizatorių, užsiimančią adresų nurodymu tinkle. Ši funkcija vadinama *NAT (Network adress translation)*.

<sup>68</sup> <http://tools.ietf.org/html/rfc2050>

<sup>69</sup> <http://tools.ietf.org/html/rfc1918>

27.2 lentelė. Vidinių adresų grupės

Adresų diapazonas	Tinklų klasė ir skaičius	Tinklo pavyzdys
10.0.0.0–10.255.255.255	Vienas A klasės tinklas	10.0.0.0/8
172.16.0.0–172.31.255.255	16 B klasės tinklų	172.22.0.0/16
192.168.0.0– 192.168.255.255	256 C klasės tinklai	192.168.222.0/24

Vidinių adresų naudojimas leidžia sumažinti reikalingų unikalų *IP* numerių skaičių ir palengvina bei atpigina tinklų kūrimą, nes vidiniai adresai naudojami nemokamai, nereikia atlikti jokių formalų veiksmų. Kai kam kyla klausimas, ar naudodami *NAT* negalėtume praplėsti *IPv4* adresų erdvės? Taip, *IPv4* leidžia taupyti *IP* adresus, tačiau tai nėra ilgalaikis adresų trūkumo sprendimas. *NAT* racionalu naudoti, kai nedidelio tinklo vartotojai jungiasi prie žiniatinklio, siunčia ir gauna laiškus bei failus. Šios paslaugos prieinamos taip, tarsi visi kompiuteriai būtų tiesiogiai prijungti prie interneto, bet jų prisijungimas nevisavertis. *NAT* apsunkina daugelį sudėtingesnių užduočių, pavyzdžiui, susijusių su srautinių duomenų perdavimu. Be to, tinklui veikiant per kelias *NAT* pakopas tampa vis sunkiau diagnozuoti kylančias ryšio problemas.

Apie 1990 m. imta rimtai ruoštis galimam adresų erdvės trūkumui ir 1992 m. aprašytas *IPv6*, kurio adresų erdvė turi didelį rezervą<sup>70</sup>. Tačiau internetas *IPv4* adresų trūkumo problemą kol kas sprendžia panaudodamas vidinius adresus, todėl perėjimas prie *IPv6* vyksta vangiai.

Numatyta ir daugiau rezervuotų *IP* adresų grupių. Paprastai tokie adresai, kaip ir čia minėti vidiniai adresai, gali būti naudojami specialioms tikslams, bet nėra matomi globaliame tinkle (27.3 lentelė). *RFC3330* ne tik išvardina rezervuotus adresus, bet ir

<sup>70</sup> <http://en.wikipedia.org/wiki/IPv6>

pateikia nuorodas į kitus *RFC* dokumentus, detalizuojančius kiekvienos rezervuotos adresų grupės naudojimą<sup>71</sup>.

27.3 lentelė. Kiti dažnai naudojami rezervuoti adresai

Adresų grupė	Paskirtis	Aprašas
0.0.0.0/8	nuoroda į save: „šis tinklas“	<i>RFC 1700</i>
127.0.0.0/8	nuoroda į save: „šis kompiuteris“	<i>RFC 1700</i>
169.254.0.0/16	supaprastintas vietinis tinklas	<i>RFC 3927</i>
192.0.2.0/24	tekste pateikiamiems pavyzdžiams	<i>RFC 3330</i>
192.88.99.0/24	sąveika tarp <i>IPv4</i> ir <i>IPv6</i>	<i>RFC 3068</i>
255.255.255.255	pranešimai visiems vietinio tinklo kompiuteriams	<i>RFC 1700</i>

## 27.6. Simboliniai domenų vardai

*IP* adresai užtikrina sistemos veikimą, tačiau nėra patogūs žmogui. Skaičių sekas nėra lengva atsiminti ir yra daugiau šansų suklysti nei rašant žodžius. Kartais serveris perkeliamas į kitą tinklą, todėl keičiasi ir jo *IP* adresas – būtų sudėtinga apie tai pranešti visiems serverio lankytojams. Todėl yra sugalvoti simboliniai domenų vardai, leidžiantys kreiptis į tinklo resursus lengviau įsi-menamu simboliu vardu. Pavyzdžiui, *lt.wikipedia.org*, o ne 91.198.174.2. Simbolinio vardo sudedamosios dalys atskiriamos tašku ir detalizuojamos iš dešinės į kairę.

Domeno vardo dalis po paskutinio taško vadinama aukščiausiojo lygio domenu. Pavyzdžiui, *.com* reiškia komercinę organizaciją, *.org* turėtų būti pelno nesiekianti organizacija. Ši vardu sistema sukurta, kai internetas kūrėsi JAV. Todėl kai kurie aukščiausiojo lygio vardai skirti JAV institucijoms: *.gov* – JAV vyriausybės organizacijos, *.mil* – JAV kariuomenė. Aukščiausiojo lygio domenų vardai suteikiami ir valstybėms. Pavyzdžiui, *.lt* (Lie-

<sup>71</sup> <http://tools.ietf.org/html/rfc3330>

tuva), *.lv* (Latvija), *.ee* (Estija). Politiniai pokyčiai ne visada lemia vardų pasikeitimus. Pavyzdžiui, yra panaikintas aukščiausio lygio domenas *.cs*, kuris buvo skirtas Čekoslovakijai, vietoj to naudojami *.cz* (Čekija) ir *.sk* (Slovakija), tačiau vis dar naudojamas domenas *.su* (*Soviet Union*), nors nebėra SSRS, o išsilaisvinusios valstybės naudoja savo domenų vardus.

Į kairę nuo aukščiausio lygio domeno iki antro taško yra antrojo lygio domeno vardas. Tokius vardus suteikia institucijos, atsakingos už aukščiausio lygio domenų vardus. Pavyzdžiui, VGTU įregistravo antrojo lygio domeną *vgtu.lt*. Aprašų pavyzdžiams yra rezervuoti šie antrojo lygio domenai: *example.com*, *example.net* ir *example.org*<sup>72</sup>.

Toliau į kairę atskirti taškais gali būti trečiojo, ketvirtojo ir žemesnių lygių domenai. Juos kuria ir prižiūri organizacija, įregistravusi pirmojo lygio domeną. Trečiojo lygio domenų vardai kartais suteikiami organizacijos padaliniams. Vardo pradžioje esantis žemiausio lygio domeno vardas paprastai nurodo paslaugą arba prisijungimo protokolą. Pavyzdžiui, *www.example.net* – žiniatinklio svetainė, o *irc.example.net* – IRC pokalbių serveris. Adresas *pop.ef.vgtu.lt* galėtų reikšti, kad tai POP3 pašto serveris Elektronikos fakultete VGTU tinkle, įregistruotame Lietuvoje.

Paketai internete adresuojami IP numeriais, tačiau tarp domenų vardų ir IP adresų nėra algoritminio ryšio, todėl būtina naudoti papildomas tarnybas, kurios tinklo mazgą vienareikšmiškai nustato pagal domeno vardą. Tokios tarnybos vadinamos DNS serveriais (*Domain Name Server*).

Duomenų transporto sistema TCP – IV lygmuo pagal OSI.

Kadangi IP protokolas paketus perduoda be sujungimo patvirtinimo, jis siekia pristatyti paketus, tačiau negarantuoja paketų pristatymo į paskirties mazgą.

Duomenų perdavimo TCP protokolu patikimumas paremtas loginiu sąveikaujančių procesų sujungimu. Kol TCP protokolo programos veikia teisingai, o sudėtinis tinklas neišsibarstęs į nesu-

<sup>72</sup> <http://tools.ietf.org/html/rfc2606>

jungiamas dalis, duomenų perdavimo klaidos *IP* protokolo lygmeniu neturės įtakos teisingiems duomenims gauti.

Protokolas *TCP* naudoja *IP* protokolą kaip transporto priemonę. Prieš išsiųsdamas duomenų blokus, *TCP* protokolas juos sudeda į *IP* paketo apvalkalą. Esant reikalui, *IP* protokolas gali šį paketą fragmentuoti ir paskui vėl sujungti, kai perdavimas vyksta per keletą tinklų bei tarpinių šliuzų.

*TCP* naudojasi *IP* protokolu ir teikia paslaugas aukštesniojo lygmens protokolo užduotims.

Šio protokolo tikslas – perduoti duomenis tarp taikomųjų procesų, vykdomų bet kuriuose tinklo mazguose. Kai *IP* protokolas pristato paketą nurodytu adresu, duomenis reikia perduoti konkrečiam procesui. Kiekvienas kompiuteris gali vykdyti keletą procesų, be to, taikomasis procesas gali turėti ir daugiau nei vieną įėjimo tašką, kuris laikomas duomenų paketo paskirties adresu.

Paketai, patenkantys į transporto lygmenį, operacinėje sistemoje organizuojami kaip daugybė eilių prie įvairių taikomųjų procesų įėjimo taškų. Tokios eilės vadinamos prievadais (angl. *ports*). Paskirties adresas, kuris naudojamas *TCP* protokole, yra taikomosios tarnybos prievado numeris. Pavyzdžiui, *FTP* tarnybai dažniausiai priskiriamas 21 prievadas, o *Telnet* tarnybai – 23. Programos kūrėjas pats gali priskirti prievado numerį, jei tas numeris nėra rezervuotas. Prievado numeris kartu su *IP* adresu vienareikšmiškai rodo taikomąjį procesą tinkle. Šis parametru rinkinys vadinamas lizdu (angl. *socket*).

*TCP* protokolo duomenų vienetas vadinamas segmentu. Informacija, patenkanti į *TCP* protokolą iš aukštesniojo lygmens protokolu, traktuojama kaip nestruktūruotas baitų srautas. Duomenys kaupiami buferyje, iš jo paskui paimama tam tikra nuosekli duomenų dalis, kuri ir vadinama segmentu.

Segmento dydis parenkamas taip, kad pakuojamas segmentas tilptų visas, t. y. maksimalus segmento dydis neturi viršyti *IP* paketo maksimalaus duomenų lauko dydžio. Kitaip reikėtų segmentą fragmentuoti, t. y. segmentą padalinti, kad jis tilptų į *IP* paketą.

Sujungimas *TCP* protokole identifikuojamas abiejų sąveikaujančių procesų lizdų pora. Formaliai sujungimą galima apibrėžti kaip parametrų rinkinį, charakterizuojantį duomenų apsikeitimą tarp dviejų procesų. Be išsamių procesų adresų, šiame rinkinyje yra ir parametrai, kurių reikšmės nustatomos per *TCP* modulį „pasi-kalbėjimą“. Tokiais parametrais gali būti segmentų dydžiai, duomenų kiekis, kurį galima perduoti negavus patvirtinimo, pradinis ir esamas perduodamų baitų numeriai. *TCP* užtikrina dviejų krypčių ryšį.

*TCP* protokole naudojamas slystančiojo lango patvirtinimų algoritmas. Tai reiškia, kad segmentų skaičius, lygus lango dydžiui, gali būti persiųstas nelaukiant patvirtinimo. Tada procesas sustoja, kol sulaukiama bent vieno patvirtinimo iš priimančiosios pusės. Jei, tarkim, sulaukta 3 segmentų gavimo patvirtinimo, langas pasislenka per 3 pozicijas – vadinasi, dar 3 segmentus galima pasiųsti, nelaukiant patvirtinimo ir t. t. Jeigu kuris segmentas persiunčiamas neteisingai, jo gavimo patvirtinimas nesiunčiamas. Taip patvirtinimo negavimas reiškia, kad arba segmentas, arba pats patvirtinimas nepasiekė adresato. Patvirtinimas siunčiamas irgi kaip segmentas, kuriame yra skaičius, vienetu didesnis už maksimalaus baito numerį gautajame segmente. Šis skaičius vadinamas eilės numeriu.

Perdavimo patikimumas užtikrinamas naudojant patvirtinimus ir eilės numerius. Kiekvienam duomenų baitui priskiriamas eilės numeris. Pirmojo duomenų baito eilės numeris segmente perduodamas kartu su šiuo segmentu ir vadinamas segmento eilės numeriu. Segmentai taip pat turi eilės numerį, kuris yra kito laukiamo duomenų baito, perduodamo priešinga kryptimi, numeris. Kai yra perduodamas segmentas su duomenimis, jo kopija pateikiama į pakartotinio siuntimo eilę ir paleidžiamas laikmatis. Gavus pristatymo patvirtinimą, segmentas pašalinamas iš eilės. Jei per nustatytą laiką patvirtinimas negaunamas, segmentas išsiunčiamas pakartotinai.

### 27.7. Perdavimo saugumas

**V lygmuo pagal OSI.** Aprašyta sistema leidžia perduoti duomenis tarp bet kurių dviejų internete veikiančių įrenginių. Duomenys perduodami per tarpinius kompiuterius, kurie dažniausiai priklauso skirtingiems savininkams. Panaudojęs specialią programinę įrangą piktavališkas asmuo galėtų perskaityti perduodamą turinį. Dar blogiau, jei neapsaugoti perduodami prisijungimo slaptažodžiai. Tuomet būtų galima įsilaužti į informacinės sistemas.

Siekiant užtikrinti saugumą kuriami šifravimo ir kitokie apsaugos protokolai. Pavyzdžiui, *SSL (Secure Sockets Layer)* – kriptografinis protokolas, skirtas šifruojant apsaugoti informacijai, sklindančiai internete. *SSL* šifruoti naudoja tiek simetrinę, tiek asimetrinę kriptografiją. Simetrinė kriptografija yra daug greitesnė už asimetrinę, todėl ji naudojama persiunčiant duomenis, taip taupomas persiuntimo laikas. Asimetrinė kriptografija naudojama tik apsikeičiant raktais. Serveris atsiunčia klientui savo viešąjį raktą, kuriuo klientas užšifruoja sugeneruotą simetrinį šifro raktą, taip jis apsaugomas nuo tarpininkų.

### 27.8. Vartotojo programos

**VII lygmuo pagal OSI.** Kiekvienas žemesnis lygmuo užtikrina aukštesniojo lygmens veikimą. Visų jų viršūnėje – VII lygmuo – vartotojo programos. Kai kuriuos paplitusius vartotojų programų sąveiką užtikrinančius protokolus aptarsime kituose skyriuose.

### 27.9. Klausimai ir uždaviniai

1. Kaip atvirieji standartai veikia tinklų plėtrą?
2. Ar gali atvirieji standartai užtikrinti saugų kompiuterių sujungimą?
3. Aprašykite kuo mažesnę namų vidinį tinklą, jei namie yra du kompiuteriai ir maršrutizatorius.

4. VGTU EF vidinis tinklas 193.219.149.0/25.
  - Kokia yra jo kaukė?
  - Kiek bitų skiriama šio tinklo numeriui, kiek bitų skiriama įrenginio numeriui? Kiek skirtingų įrenginių gali turėti unikalius *IP* šiame tinkle?
  - Koks turėtų būti šio tinklo pranešimų transliavimo (*broadcast*) adresas?
  - Pateikite šio tinklo *IP* pavyzdžių.
5. Raskite penkis naujausius *RFC* dokumentus. Apie ką juose kalbama?