

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Skaičiavimai laiko logikos fragmentams

Calculi for fragments of temporal logics

Magistro baigiamasis darbas

Atliko:	Lukas Maksimiak	(parašas)
Darbo vadovas:	asist. dr. Adomas Birštunas	(parašas)
Recenzentas:	asist. dr. Julius Andrikonis	(parašas)

Vilnius – 2025

Santrauka

Klasikinę logiką praplėtus modalumo operatoriais gaunama modalumo logika. Laiko logika gaunama klasikinę logiką papildžius laiko logikos modaliniais operatoriais. Šiame darbe nagrinėjama tiesinė laiko logika. Magistro baigiamojo darbo tikslas – išskirti tiesinės laiko logikos fragmentus ir jiems sukonstruoti efektyvius sekvencinius skaičiavimus. Plačiausiai žinomi trys pagrindiniai sekvenciniai skaičiavimai tiesinei laiko logikai: begalinis, invariantinės taisyklės ir ciklinio tipo. Šio darbo pagrindiniai rezultatai yra apibrėžtas unarinis tiesinės laiko logikos fragmentas, apibrėžtos ir įrodytos pagalbinės lemos tiesinės laiko logikos sekvenciniam skaičiavimui. Sukurtas naujas tiesinės laiko logikos sekvencinis skaičiavimas, nenaudojantis ciklą aptikimo mechanizmo išvedimo paieškoje, kuris apibrėžtame fragmente yra efektyvesnis už klasikinį sekvencinį skaičiavimą. Įrodytas ekvivalentumas tarp naujai sukurtų skaičiavimų ir klasikinio tiesinės laiko logikos sekvencinio skaičiavimo apibrėžtame fragmente.

Raktiniai žodžiai: sekvenciniai skaičiavimai, tiesinė laiko logika, laiko logikos fragmentai, modalinė logika, ciklai

Summary

Modal logic is obtained by extending classical logic with modality operators. Temporal logic is obtained by extending classical logic with temporal operators. This paper deals with linear temporal logic. The work objective is to distinguish fragments of linear time logic and construct efficient sequential calculi for them. There are three most widely known sequent calculi for linear temporal logic: infinite, invariant rule, and cyclic type. The main results of this work are the definition of a unary fragment of linear temporal logic, and the definition and proof of auxiliary lemmas for sequent calculus in linear temporal logic. A new sequent calculus for linear temporal logic has been developed, which does not use cycle detection in proof search and is more efficient than the classical sequent calculus in the defined fragment. The equivalence between the newly developed sequent calculi and the classical sequent calculus for linear temporal logic in the defined fragment has been proved.

Keywords: sequent calculi, linear temporal logics, fragments of LTL, modal logics, cycles

Turinys

Išvadas	4
1. Esminės matematinės logikos sąvokos	6
2. Teiginių logika	7
2.1. Loginės operacijos ir formulės	7
2.2. Skaičiavimai	7
2.2.1. Hilberto tipo teiginių skaičiavimas	7
2.2.2. Sekvencinis skaičiavimas	8
3. Modalumo logikos	11
3.1. Semantika	11
3.2. Modalumo logikų skaičiavimai	13
3.2.1. Hilberto tipo skaičiavimai modalumo logikoms	13
3.2.2. Sekvenciniai skaičiavimai modalumo logikoms	14
3.3. Laiko logikos	16
3.3.1. Tiesinė laiko logika	16
3.3.2. Skaidaus laiko logika	17
3.3.3. Žinomi sekvenciniai skaičiavimai tiesinei laiko logikai	18
4. Laiko logikos pritaikymo sferos	21
4.1. Laikinosios duomenų bazės	21
4.2. Žinių reprezentacija ir samprotavimas	21
4.3. Multi-agentų sistemos	21
4.4. Veiksmų laiko logika	22
5. Fragmentai	23
5.1. Naujo fragmento apibrėžimas	23
6. <i>LTL</i> skaičiavimo savybės	25
7. Skaičiavimas <i>LTL1</i>	30
7.1. Sekvencinio skaičiavimo <i>LTL1</i> apibrėžimas	30
7.2. <i>LTL1</i> ir klasikinio sekvencinio skaičiavimo ekvivalentumas	34
7.3. <i>LTL1</i> ir <i>LTL</i> skaičiavimų palyginimas	36
8. Sekvenciniai skaičiavimai su apribotomis taisyklėmis	38
8.1. Skaičiavimas <i>LTL2</i>	38
8.1.1. <i>LTL</i> ir <i>LTL2</i> ekvivalentumas	39
8.2. Skaičiavimas <i>LTL3</i>	40
8.2.1. <i>LTL1</i> ir <i>LTL3</i> ekvivalentumas	40
8.2.2. Skaičiavimo <i>LTL3</i> savybės	42
9. Skaičiavimas <i>LTL4</i>	44
9.1. <i>LTL3</i> ir <i>LTL4</i> ekvivalentumas	44
9.2. <i>LTL</i> ir <i>LTL4</i> ekvivalentumas	44
9.3. <i>LTL4</i> ir <i>LTL</i> skaičiavimų palyginimas	45
10. Rezultatai ir išvados	46
Literatūra	48

Ivadas

Logika yra mokslas apie dėsnius, nuo kurių priklauso taisyklingas mąstymas. Ši mokslo šaka nagrinėja žmogaus mąstymo formą. Logika, kaip savarankiška mokslo šaka susiformavo 20 a. ketvirtajame dešimtmetyje. Matematinė logika remiasi matematika ir tyrinėja matematinius samprotavimus. Matematikoje, pagrindinis tyrinėjimo metodas yra laikomas įrodymu, o ne eksperimentu, todėl svarbus uždavinys yra rasti tam tikras logikos taisykles, kuriomis naudojantis galima būtų įrodyti teoremas.

Teiginių ir predikatų logikos yra matematinės logikos atšakos, kuriose formaliosios kalbos simboliais bei loginėmis operacijomis nusakomi teiginiai ar predikatai. Teiginiu vadiname bet kokią sakinį, kuris yra teisingas arba klaidingas. Predikatu vadiname teiginį, kuris savyje turi tam tikrą objektą (ar objektus) ir požymį, kuris nusako ar tam tikru būdu apibūdina šį objektą. Šios logikos yra labai naudingos, nes jos leidžia formalizuoti kalbą ir apibrėžti logikos taisykles, kurios yra labai svarbios įvairių matematikos ir informatikos sričių taikymuose.

Modalumo logikos suformuojamos klasikinės logikos kalbą praplėtus modalumo operatoriais, kurie nusako būtinumą ir galimybę. Šių logikų pagalba galime formalizuoti įvairius įsitikinimus ir abejones, o tai labiau atitinka realų pasaulį.

Tiesinė ir skaidaus laiko logikos yra specifinės modalumo logikos nagrinėjančios objektų ar reikšmių būsenas kintančias laike. Abi šios logikos skirtos formalizuoti teiginius apie objektus laiko eigoje. Tiesinėje laiko logikoje nagrinėjama vientisa laiko linija, o skaidaus laiko logikoje įmanomas laiko išsišakojimas.

Tam tikrais atvejais, norint įrodyti išsprendžiamumą, baigtinumą, ar egzistuojant dideliame sudėtingumui, nagrinėjami logikos fragmentai. Logikos fragmentas yra sintaksiškai suvaržytas visos logikos formulių poaibis. Nagrinėjant tik konkrečius logikos fragmentus lengviau jiems sukonstruoti efektyvesnius skaičiavimus, įrodyti fragmentų išsprendžiamumą ar baigtinumą. Šiame darbe bus nagrinėjama tiesinė laiko logika.

Skaičiavimai logikose leidžia nustatyti ar tam tikri teiginiai nagrinėjamoje logikoje yra teisingi. Norint įrodyti teiginio teisingumą pasitelkiamos tam tikros taisyklės, kurias pritaikant galima išvesti teiginį nagrinėjamoje logikoje. Šie skaičiavimai turi pasižymėti šiomis pagrindinėmis taisyklėmis: būti neprieštaringi (negalima išvesti teiginio ir jam priešingo teiginio) ir būti pilni (skaičiavime galima išvesti visus tapačiai teisingus teiginius ir tik tokius). Dažnai nagrinėjami ir skaičiavimai, kurie yra pilni ne tik tapačiai teisingų formulių atžvilgiu, tačiau ir kitų formulių aibių atžvilgiu. Tarkime, skaičiavime, kuris yra pilnas įvykdomų formulių atžvilgiu, galima įrodyti visas įvykdomas ir tik įvykdomas formules. Paprastai yra siekiama, kad skaičiavimai pasižymėtų pilnumo ir korektiškumo savybėmis, nes norima sukonstruoti skaičiavimą tinkamą tikrinti ar formulę yra tapačiai teisinga.

Šiame darbe nagrinėjami skaičiavimai tiesinės laiko logikos fragmentams, norint sukonstruoti efektyvesnius sekvencinius skaičiavimus. Plačiausiai žinomi sekvenciniai skaičiavimai tiesinėje laiko logikoje yra invariantinės taisyklės, begalinis ir ciklinio tipo. Šie sekvenciniai skaičiavimai skiriasi būtinumo dešinėje taisykle, kuri kiekvieno skaičiavimo atveju turi savų plusų ir minusų. Invariantinis skaičiavimas reikalauja naujos formulės įvedimo, o begaliniame skaičiavime atsiranda

begalinis kiekis sekvencijų, todėl šie skaičiavimai yra arba labai sunkiai arba praktiškai neautomatizuojami.

Šį darbą sudaro 10 skyrių. Pirmajame skyriuje apibrėžiamos esminės matematinės logikos sąvokos, kurios naudojamos tolimesniuose skyriuose. Antrajame skyriuje aprašoma teiginių logika ir skaičiavimai šiai logikai. Trečiajame skyriuje nagrinėjamos modalumo logikos, jų semantika ir skaičiavimai. Ketvirtajame skyriuje aprašomos laiko logikos pritaikymo sferos. Penktajame skyriuje apibrėžiama fragmento sąvoka ir naujas fragmentas α_1 . Šestajame skyriuje įrodomos pagalbinės lemos skaičiavimui LTL . Septintajame skyriuje pateikiamas naujas skaičiavimas $LTL1$, įrodomas naujojo skaičiavimo ir klasikinio tiesinės laiko logikos sekvencinio skaičiavimo ekvivalencijos fragmente, skaičiavimai palyginami. Aštuntajame skyriuje apibrėžiamas skaičiavimas $LTL2$, įrodomas ekvivalencijos tarp LTL ir $LTL2$. Taip pat apibrėžiamas skaičiavimas $LTL3$, įrodomos specifinės skaičiavimo savybės, bei ekvivalencijos tarp $LTL1$ ir $LTL3$ fragmente α_1 . Devintajame skyriuje pateikiamas skaičiavimas beciklis skaičiavimas $LTL4$, įrodomas ekvivalencijos tarp LTL ir $LTL4$ fragmente α_1 . Galiausiai pateikiami pagrindiniai darbo rezultatai ir išvados.

Magistro baigiamojo **darbo tikslas** – išskirti tiesinės laiko logikos fragmentus ir jiems sukonstruoti efektyvius sekvencinius skaičiavimus. Magistro baigiamojo darbo metu apibrėžtam tiesinės laiko logikos fragmentui buvo sukurti sekvenciniai skaičiavimai $LTL1$, $LTL2$, $LTL3$ ir $LTL4$. Skaičiavimuose $LTL1$, $LTL3$ nebelieka ciklinių aksiomų, o skaičiavimas $LTL4$ yra visai beciklis skaičiavimas, kuriam ciklų aptikimo mechanizmo nebereikia.

1. Esminės matematinės logikos sąvokos

Šiame skyriuje aprašomos pagrindinės matematinės logikos sąvokos ir apibrėžimai, kurie vėlesniuose skyriuose papildomi arba specifikuojami.

1 Apibrėžimas. *Loginis kintamasis yra kintamasis galintis įgyti tik dvi reikšmes tiesa arba netiesa.*

2 Apibrėžimas. *Formulės apibrėžimas.*

1. *Loginis kintamasis yra formulė*
2. *Jeigu F yra formulė, tai $\odot F$ taip pat formulė, kur $\odot$ - kuris nors unarinis operatorius*
3. *Jei F, G yra formulės, tai $F \odot G$, taip pat yra formulė, kur $\odot$ - kuri nors loginė operacija ar binarinis operatorius*

Matematinėje logikoje formulės yra klasifikuojamos į tris kategorijas. Bet kuri formulė gali būti tapčiai teisinga, tapčiai klaidinga arba įvykdoma. Priklausomai nuo skirtingos logikos (teiginių, predikatų, modalumo) interpretacija yra apibrėžiama skirtingai. Toliau pateikiamas formulių klasifikavimo apibrėžimas.

3 Apibrėžimas. *Formulių klasifikavimas.*

1. *Formulė F vadinama tapčiai teisinga, jei ji teisinga su bet kuria interpretacija.*
2. *Formulė F vadinama tapčiai klaidinga, jei ji klaidinga su bet kuria interpretacija.*
3. *Formulė F vadinama įvykdoma, jei egzistuoja interpretacija su kuria formulė teisinga.*

4 Apibrėžimas. *Poformulio apibrėžimas.*

1. *Formulė F yra F poformulis.*
2. *Jei $F = \odot G$, tai formulė G ir visi G poformuliai yra ir F poformuliai, kur $\odot$ kuris nors unarinis operatorius.*
3. *Jei $F = G \odot H$, tai formulės G, H bei jų poformuliai yra ir F poformuliai, kur $\odot$ - kuris nors binarinis operatorius*

2. Teiginių logika

Teiginiu vadiname sakinį, kurio atžvilgiu prasmingas klausimas teisingas jis ar klaidingas [Nor04]. Teiginius žymime mažosiomis lotyniškėmis raidėmis pvz. p, q, r, s . Jeigu teiginys yra teisingas, sakoma, jog jo reikšmė yra lygi t , jeigu teiginys klaidingas, sakoma, kad jo reikšmė lygi k . Informatikos moksle patogiu teiginių kitimo sritį pakeisti į aibę $\{0,1\}$, t.y. k keičiamas į 0 , o t į 1 . Kadangi teiginiai gali įgauti dvi skirtingas reikšmes, juos vadiname loginiais kintamaisiais. Sudėtinį teiginį gauname logines operacijas pritaikę teiginiams, o jo reikšmė priklauso nuo jį sudarančių paprastesnių teiginių reikšmių. Sudėtinius teiginius galime išreikšti formulėmis, kurios šiame darbe bus žymimos didžiosiomis raidėmis $F, G, H, \dots$, kurios gali būti indeksuojamos.

2.1. Loginės operacijos ir formulės

Loginių teiginių manipuliavimui yra apibrėžtos tam tikros loginės operacijos, tokios kaip neigimas, disjunkcija, konjunkcija, implikacija ar ekvivalentumas. Kiekvienai teiginių logikos formulei galima rasti ekvivalenčią, kurioje yra tik loginės operacijos iš aibės $\{\neg, \vee, \wedge\}$. Teiginių logikoje naudojamas unarinis operatorius $\neg$ ir binariniai operatoriai $\{\vee, \wedge, \rightarrow\}$.

5 Apibrėžimas. *Loginių kintamųjų aibės $\{p_1, \dots, p_n\}$ interpretacija teiginių logikoje vadiname kurią nors funkciją v su apibrėžimo sritimi $\{p_1, \dots, p_n\}$ ir reikšmėmis $\{1, 0\}$.*

Formulių klasifikavimo apibrėžimas toks pat kaip (3) apibrėžime, kur interpretacija apibrėžiama kaip (5) apibrėžime.

2.2. Skaičiavimai

Šiame skyriuje aprašomi dažniausiai literatūroje naudojami teiginių logikos skaičiavimai. Skaičiavimu nusakome įrodomų jame formulių aibę [Nor04].

2.2.1. Hilberto tipo teiginių skaičiavimas

Šiame skyrelyje aprašomas pataisytas ir papildytas konjunkcijos ir disjunkcijos aksiomomis G. Frege 1879 m. aprašytas skaičiavimas. Tokio tipo skaičiavimai įprastai vadinami **Hilberto tipo** skaičiavimai, kadangi vokiečių matematikas D. Hilbertas taip pat nagrinėjo skaičiavimus ir šioje srityje pasiekė gerų rezultatų [Nor04].

Skaičiavimas yra nusakomas aksiomomis, kur A, B ir C bet kokios teiginių logikos formulės:

1. $A \rightarrow (B \rightarrow A)$
2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
2. $(A \wedge B) \rightarrow A$
2. $(A \wedge B) \rightarrow B$
3. $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$

3.
 1. $A \rightarrow (A \vee B)$
 2. $B \rightarrow (A \vee B)$
 3. $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$
4.
 1. $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$
 2. $A \rightarrow \neg\neg A$
 3. $\neg\neg A \rightarrow A$

ir taisykle *Modus Ponens (MP)*:

$$(MP) \quad \frac{A \quad A \rightarrow B}{B}$$

1.1 - 4.3 vadinamos aksiomų schemomis, o pakeitus A, B, C į konkrečias formules yra gaunamos aksiomos.

6 Apibrėžimas. Įrodymu Hilberto tipo teiginių skaičiavime vadiname baigtinę formulių seką, kurioje kiekviena iš formulių yra gauta iš prieš ją esančių formulių pagal *modus ponens* taisyklę arba yra aksioma.

1 Teiginys. Formulė F yra įrodoma Hilberto tipo teiginių skaičiavime tada ir tik tada, kai F yra tapaciai teisinga.

2.2.2. Sekvencinis skaičiavimas

Vokiečių logikas G. Gentzen 1930 m. aprašė sekvencinį skaičiavimą, kuriame išvedimo paieška gali būti automatizuojama ir daugeliu atveju paprastesnė nei Hilberto tipo skaičiavime. Sekvencinis skaičiavimas įgalina formulių teisingumo įrodymą.

7 Apibrėžimas. Sekvencija vadinamas reiškinyss pavidalo

$F_1, \dots, F_n \vdash G_1, \dots, G_m$, kur F_i ($i = 1, \dots, n$) ir G_i ($i = 1, \dots, m$) yra formulės, o $n + m > 0$.

Sekvencijoje $\Gamma \vdash \Delta$ seka Γ vadinama antecedentu, o Δ vadinama sukcedentu.

Aksiomos: $\Gamma, F \vdash \Delta, F$.

Struktūrinės taisyklės:

$$(\text{silpninimas}) \quad \frac{\Gamma \vdash \Delta}{F, \Gamma \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, F}$$

$$(\text{prastinimas}) \quad \frac{F, F, \Gamma \vdash \Delta}{F, \Gamma \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta, F, F}{\Gamma \vdash \Delta, F}$$

$$(\text{perstatymas}) \quad \frac{\Gamma_1, F, G, \Gamma_2 \vdash \Delta}{\Gamma_1, G, F, \Gamma_2 \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta_1, F, G, \Delta_2}{\Gamma \vdash \Delta_1, G, F, \Delta_2}$$

Loginių operacijų taisyklės:

$$(\neg \vdash) \quad \frac{\Gamma \vdash \Delta, F}{\neg F, \Gamma \vdash \Delta}, \quad (\vdash \neg) \quad \frac{F, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg F}$$

$$(\wedge \vdash) \quad \frac{F, G, \Gamma \vdash \Delta}{F \wedge G, \Gamma \vdash \Delta}, \quad (\vdash \wedge) \quad \frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \wedge G}$$

$$(\vee \vdash) \quad \frac{F, \Gamma \vdash \Delta \quad G, \Gamma \vdash \Delta}{F \vee G, \Gamma \vdash \Delta}, \quad (\vdash \vee) \quad \frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}$$

$$(\rightarrow \vdash) \quad \frac{\Gamma \vdash \Delta, F \quad G, \Gamma \vdash \Delta}{F \rightarrow G, \Gamma \vdash \Delta}, \quad (\vdash \rightarrow) \quad \frac{F, \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \rightarrow G}$$

8 Apibrėžimas. Medžio pavidalo orientuotą grafą, kurio visos viršūnės pažymėtos sekvencijomis (šaknis - pradinė sekvencija) ir kiekviena viršūnė (išskyrus lapus) gauta iš tiesiogiai virš jos esančių sekvencijų, pritaikius kurią nors sekvencinio skaičiavimo taisyklę, vadiname išvedimo paieškos medžiu.

9 Apibrėžimas. Sakome, kad formulė F yra išvedama sekvenciniame skaičiavime, kai egzistuoja bent vienas išvedimo paieškos medis, kurio visuose lapuose yra aksiomos, o medžio šaknyje yra sekvencija $\vdash F$.

10 Apibrėžimas. Sakome, kad taisyklė T apverčiama, jei jos prielaidos išvedamos sekvenciniame skaičiavime tada ir tik tada, kai išvedama išvada.

11 Apibrėžimas. Taisyklė vadinama pusiau apverčiama, jeigu egzistuoja tokia taisyklės prielaida, kuri yra išvedama, kai išvedama taisyklės išvada.

12 Apibrėžimas. *OR taisykle vadiname sekvencinio skaičiavimo taisyklę pavidalo:*

$$(T) \frac{\Pi_1 \vdash \Delta_1 \quad || \quad \Pi_2 \vdash \Delta_2 \quad || \quad \dots \quad || \quad \Pi_n \vdash \Delta_n}{\Gamma \vdash \Sigma}$$

Šio pavidalo taisyklė reiškia, jog taisyklės išvada išvedama, jeigu išvedama bent viena iš OR taisyklės prielaidų.

Kaip ir kitur literatūroje [Bir07] šiame darbe naudojama OR tipo taisyklė. OR tipo taisyklė yra pusiau apverčiama. Norint parodyti, kad sekvencija yra išvedama, reikia sukonstruoti išvedimo medį, kuriame kiekviename OR taisyklės taikyme yra palikta bent viena OR taisyklės prielaida ir visos medžio šakos baigiasi aksiomomis. Dažnai konstruojant išvedimo paieškos medį tenka patikrinti visas OR taisyklės prielaidas, norint rasti tokią, kuri yra išvedama, arba norint parodyti, kad tokios nėra.

2 Teiginys. *Formulė F yra tapačiai teisinga tada ir tik tada, kai sekvencija $\vdash F$ išvedama klasi-
kiniame teiginių logikos sekvenciniame skaičiavime.*

Pavyzdys. Klasikiniu teiginių logikos sekvenciniu skaičiavimu išvesime sekvenciją
 $S : (F \wedge G) \rightarrow H, F \wedge \neg H \vdash \neg G$.

$$\frac{\frac{F, \neg H \vdash \neg G, F \quad (\vdash \neg) \frac{F, \neg H, G \vdash G}{F, \neg H \vdash \neg G, G}}{(\vdash \rightarrow) \frac{F, \neg H \vdash \neg G, F \wedge G}{(F \wedge G) \rightarrow H, F, \neg H \vdash \neg G}} \quad (\neg \vdash) \frac{H, F \vdash \neg G, H}{H, F, \neg H \vdash \neg G}}{(\wedge \vdash) \frac{(F \wedge G) \rightarrow H, F, \neg H \vdash \neg G}{(F \wedge G) \rightarrow H, F \wedge \neg H \vdash \neg G}}$$

Sekvenciniu skaičiavimu sėkmingai išvedėme sekvenciją S (kiekviename išvedimo paieškos medžio lape gavome aksiomas), tai reiškia, jog iš formulių esančių antecedente seka bent viena iš formulių esančių sukcedente. Pažymint sekvencijos antecedente esančias formules sujungtas konjunkcija - Θ ir sukcedente esančias formules sujungtas disjunkcija - Υ gauname, jog formulė $\Theta \rightarrow \Upsilon$ yra tapačiai teisinga.

3. Modalumo logikos

Klasikinės logikos kalbą praplėtę modalumo operatoriais, nusakančiais būtinumą ir galimybę, gauname modalumo logiką. Šių operatorių dėka galime formalizuoti tvirtinimus, kurie reiškia tvirtą įsitikinimą, kurio nors teiginio teisingumu ar klaidingumu arba abejones, žinojimą, tikėjimą ar dar kitokį samprotavimuose naudojamą aspektą. Viena iš pagrindinių priežasčių, lėmusiu modalumo logikos atsiradimą, yra noras išvengti implikacijos paradokso – iš klaidingo teiginio galimas bet kokio teiginio išvedimas [Nor04]. Šiuo metu yra sugalvota daug skirtingų modalumo logikų variantų, žinomiausios jų: K, T, S4, S5, KT45 ir laiko logikos. Skirtingas modalumo logikas charakterizuoja tam tikra skirtinga aksiomų aibė. Taip pat, kai konkrečiuose aksiomose yra specifinių pasaulių sąryšio apribojimų, pavyzdžiui, logikoje S4, formulė yra tapačiai teisinga tada ir tik tada, kai ji teisinga kiekvienos struktūros, kurios sąryšis refleksyvus ir tranzityvus, bet kuriame pasaulyje. Kokias tiksliai aksiomas reikia pridėti prie teiginių skaičiavimo, kad būtų sukurta tinkama modalinės logikos sistema, yra filosofinės nuomonės klausimas, dažniausiai nulemtas teoremų, kurias norima įrodyti [Nor04].

3.1. Semantika

Modalumo logikoje būtinumo operatorius žymimas $\Box$, o galimumo $\Diamond$. Formulės apibrėžimas toks pats kaip 2 apibrėžime, tik 2-ame punkte $\odot \in \{\neg, \Box, \Diamond\}$ ir 3-ame punkte $\odot \in \{\vee, \wedge\}$.

Paprastumo dėlei, dažnai nagrinėjamos formulės, kuriose yra tik vienas modalumo operatorius $\Box$. Tokiu atveju formulė $\Diamond F$ yra žymima $\neg\Box\neg F$.

13 Apibrėžimas. F yra be-modulinė formulė, jeigu šioje formulėje nėra nei vieno $\Box$ modalumo operatoriaus.

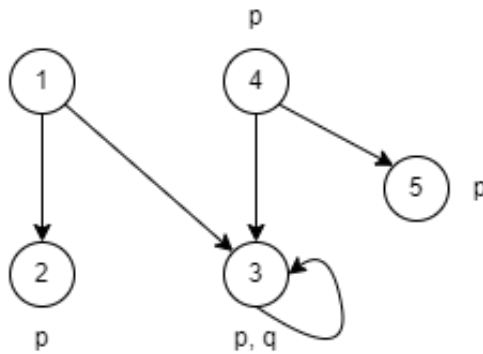
14 Apibrėžimas. Teiginių modalumo logikų formulės F Kripke struktūra vadiname trejetą $\Phi = (S, R, V)$, kur S – kuri nors netuščia aibė, vadinama galimų pasaulių aibė, R – apibrėžtas aibėje S binarusis predikatas, vadinamas pasaulių sąryšiu, V – aibė interpretacijų pasauliuose.

Tegul Φ yra tam tikros formulės F , kuri nors struktūra. Tvirtinimą, jog formulė teisinga struktūros Φ pasaulyje β , suvokiame, kad formulė teisinga su pasaulį β atitinkančia interpretacija iš V [Nor04]. Formulės F teisingumas struktūros Φ pasaulyje β nusakomas taikant indukciją pagal formulės pavidalą. Kripke struktūrą taip pat galime atvaizduoti grafu.

15 Apibrėžimas. Formulės teisingumo apibrėžimas.

- jei F yra loginis kintamasis, tai F yra teisinga tada ir tikrai tada, kai šis loginis kintamasis yra teisingas pasaulyje β
- jei $F = \neg G$, tai F teisinga tada ir tikrai tada, kai G yra klaidinga pasaulyje β
- jei $F = G \wedge H$, tai F teisinga tada ir tikrai tada, kai abi G, H teisingos pasaulyje β
- jei $F = G \vee H$, tai F teisinga tada ir tikrai tada, kai bent viena iš G, H teisingos pasaulyje β
- jei $F = \Box G$, tai F teisinga tada ir tikrai tada, kai G teisinga visuose tokiuose pasauliuose β' , kuriems $R(\beta, \beta') = t$.
- jei $F = \Diamond G$, tai F teisinga tada ir tikrai tada, kai yra bent vienas toks pasaulis β' , kuriam $R(\beta, \beta') = t$ ir G teisinga pasaulyje β' .

Pavyzdys. Imkime Kripke struktūrą $\Phi_* = (M, R, V)$, kurioje nagrinėjami du loginiai kintamieji p ir q , kur $M = \{1, 2, 3, 4, 5\}$, $R = \{(1,2), (1,3), (4,3), (4,5)\}$ ir $V = \{(1, 0, 0), (2, 1, 0), (3, 1, 1), (4, 1, 0), (5, 1, 0)\}$. Aibės V elementai yra sudaryti iš trejeto, kur pirmas elementas nusako pasaulį, antrasis ir trečiasis atitinkamai p ir q loginių kintamųjų interpretaciją. Grafe prie atitinkamų pasaulių pažymimi loginiai kintamieji įgyjantys reikšmę 1. Orientuotas grafas apibūdinantis šią struktūrą 1.



1 pav. Kripke struktūrą Φ_* apibūdinantis grafas.

Toliau galime svarstyti apie tam tikrų formulių teisingumą specifiniuose pasauliuose. Pastebime, jog formulė $\Diamond q$ yra neteisinga pasaulyje 5 (žymime $\Phi_*, 5 \not\models \Diamond q$), tačiau teisinga pasaulyje 1 (žymime $\Phi_*, 1 \models \Diamond q$).

16 Apibrėžimas. Formalus modalumo logikos formulių klasifikavimas.

- Sakoma, kad formulė F įvykdoma, jei egzistuoja tokia struktūra $\Phi = (M, R, V)$ ir pasaulis $\beta \in M$, kad F teisinga pasaulyje β
- Sakoma, kad formulė F tapaciai teisinga, jei ji teisinga bet kurios struktūros kiekviename pasaulyje
- Sakoma, kad formulė F tapaciai klaidinga, jei ji klaidinga bet kurios struktūros kiekviename pasaulyje

3.2. Modalumo logikų skaičiavimai

Šiame skyriuje aprašomi dažniausiai naudojami skaičiavimai modalumo logikoms. Dažniausiai šių skaičiavimų pagrindą sudaro klasikinės teiginių logikos taisyklės ir papildomai apibrėžiamos taisyklės modalumo operatoriams.

3.2.1. Hilberto tipo skaičiavimai modalumo logikoms

Hilberto tipo skaičiavimas modalumo logikai gaunamas klasikinį skaičiavimą su aksiomomis 1.1 - 4.3 papildžius aksiomomis:

k. $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$

t. $\Box A \rightarrow A$

4. $\Box A \rightarrow \Box \Box A$

5. $\Diamond A \rightarrow \Box \Diamond A$

d. $\Box A \rightarrow \Diamond A$

bei taisykle:

$$\frac{A}{\Box A}$$

Skirtingos modalumo logikos yra nusakomos skirtingais skaičiavimais. Modalumo logikos skaičiavimas, kuriame yra 1.1 - 4.3 aksiomos ir

- k aksioma, vadinamas K logika
- k, t aksiomos, vadinamas T logika
- k, t, 4 aksiomos, vadinamas S4 logika
- k, t, 5 aksiomos, vadinamas S5 logika
- k, d aksiomos, vadinamas D logika
- k, t, 4, 5, vadinamas KT45 logika

3.2.2. Sekvenciniai skaičiavimai modalumo logikoms

Šiame skyrelyje aprašomi sekvenciniai skaičiavimai gerai žinomoms modalumo logikoms K, T, S4. Šiuose sekvenciniuose skaičiavimuose yra visos loginės taisyklės nurodytos 2.2.2 ir papildytos atitinkamomis taisyklėmis modalumo operatoriui $\Box$.

Modalumo logika **K**:

$$(\vdash \Box) \quad \frac{\Gamma \vdash F}{\Sigma, \Box \Gamma \vdash \Delta, \Box F}$$

Modalumo logika **T**:

$$(\Box \vdash) \quad \frac{F, \Box F, \Pi \vdash \Delta}{\Box F, \Pi \vdash \Delta} \qquad (\vdash \Box) \quad \frac{\Gamma^* \vdash F}{\Sigma, \Box \Gamma \vdash \Delta, \Box F}$$

Modalumo logika **S4**:

$$(\Box \vdash) \quad \frac{F, \Box F, \Pi \vdash \Delta}{\Box F, \Pi \vdash \Delta} \qquad (\vdash \Box) \quad \frac{\Box \Gamma \vdash F}{\Sigma, \Box \Gamma \vdash \Delta, \Box F}$$

Modalumo logika **S5**:

$$(\Box \vdash) \quad \frac{F, \Pi \vdash \Delta}{\Box F, \Pi \vdash \Delta} \qquad (\vdash \Box) \quad \frac{\Box \Gamma \vdash F, \Box \Delta}{\Sigma, \Box \Gamma \vdash \Box \Delta, \Box F}$$

Modalumo logika **KD45**:

$$(\Box) \quad \frac{\Box \Gamma_1, \Gamma_2 \vdash \Box \Delta, F, \Box F}{\Pi_1, \Box \Gamma_1, \Box \Gamma_2 \vdash \Box \Delta, \Box F, \Pi_2}$$

Čia Π, Δ yra baigtinės formulų sekos; Σ - baigtinė seka formulų, neprasidedančių operatori $\Box$; $\Box \Gamma$ - baigtinė seka formulų, prasidedančių operatoriumi $\Box$; Γ - gauta iš $\Box \Gamma$, išbraukus iš visų $\Box \Gamma$ formulų pirmąsias $\Box$ įeitis; F - formulė [Nor04].

Pavyzdys. [Nor04]. Išvesime sekvenciją $\vdash \neg \Box \neg (F \vee \Box \neg F)$ logikos S4 sekvenciniame skaičiavime.

$$\begin{array}{c}
 (\vdash \vee) \frac{F, \Box \neg (F \vee \Box \neg F) \vdash F, \Box \neg F}{F, \Box \neg (F \vee \Box \neg F) \vdash F \vee \Box \neg F} \\
 (\neg \vdash) \frac{F, \neg (F \vee \Box \neg F), \Box \neg (F \vee \Box \neg F) \vdash}{F, \Box \neg (F \vee \Box \neg F) \vdash} \\
 (\Box \vdash) \frac{F, \Box \neg (F \vee \Box \neg F) \vdash}{\Box \neg (F \vee \Box \neg F) \vdash \neg F} \\
 (\vdash \neg) \frac{\Box \neg (F \vee \Box \neg F) \vdash \neg F}{\Box \neg (F \vee \Box \neg F) \vdash F, \Box \neg F} \\
 (\vdash \Box) \frac{\Box \neg (F \vee \Box \neg F) \vdash F, \Box \neg F}{\Box \neg (F \vee \Box \neg F) \vdash F \vee \Box \neg F} \\
 (\neg \vdash) \frac{\Box \neg (F \vee \Box \neg F) \vdash F \vee \Box \neg F}{\neg (F \vee \Box \neg F), \Box \neg (F \vee \Box \neg F) \vdash} \\
 (\Box \vdash) \frac{\neg (F \vee \Box \neg F), \Box \neg (F \vee \Box \neg F) \vdash}{\Box \neg (F \vee \Box \neg F) \vdash} \\
 (\vdash \neg) \frac{\Box \neg (F \vee \Box \neg F) \vdash}{\vdash \neg \Box \neg (F \vee \Box \neg F)}
 \end{array}$$

Pastebėkime, jog taisyklė $(\Box \vdash)$ pakartoja formulę su $\Box$ operatoriumi. Jeigu šioje taisyklėje to nebūtų, minėtoji sekvencija būtų neišvedama:

$$\begin{array}{c}
 (\vdash \neg) \frac{F \vdash}{\vdash \neg F} \\
 (\vdash \Box) \frac{\vdash \neg F}{\vdash F, \Box \neg F} \\
 (\vdash \vee) \frac{\vdash F, \Box \neg F}{\vdash F \vee \Box \neg F} \\
 (\neg \vdash) \frac{\vdash F \vee \Box \neg F}{\neg (F \vee \Box \neg F) \vdash} \\
 (\Box \vdash) \frac{\neg (F \vee \Box \neg F) \vdash}{\Box \neg (F \vee \Box \neg F) \vdash} \\
 (\vdash \neg) \frac{\Box \neg (F \vee \Box \neg F) \vdash}{\vdash \neg \Box \neg (F \vee \Box \neg F)}
 \end{array}$$

3.3. Laiko logikos

Formalizuodami uždavinius klasikinėmis logikos formulėmis, atskleidžiame nagrinėjamų objektų ar reikšmių būsenas. Keičiantis laikui, kinta objektų vertės, todėl formalizavimui reikia kitokios logikos, kuri atsižvelgtų į laiko faktorių [Nor04]. Toliau šiame skyrelyje nagrinėsime laiko logiką, kurios modelis yra baigtiniai orientuoti grafai.

Formulės apibrėžimas toks pats kaip klasikinės teiginių logikos formulės apibrėžimas, tik papildomas

- Jei F formulė, tai $\odot F$ taip pat formulė, kur $\odot \in \{\circ, \diamond, \square\}$
- Jei F, G formulės, tai $\odot(FUG)$ taip pat formulė, kur $\odot \in \{A, E, \emptyset\}$

17 Apibrėžimas. Laiko logikos Kripke struktūra vadiname ketvertą $\Phi = (S, I, R, L)$ kur S – kuri nors baigtinė aibė, vadinama būsenų aibe, $I \subseteq S$ – netuščia aibė, vadinama pradinių būsenų aibe, $R \subseteq S \times S$ – perėjimų sąryšio aibė, $L : S \times 2^V$ – loginių kintamųjų interpretacijų aibė. Čia V žymi baigtinę loginių kintamųjų aibę.

Kripke struktūra yra atvaizduojama baigtiniu orientuotu grafu, kurio viršūnės atitinka būsenas, kurios reiškia skirtingus laiko momentus, o lankai perėjimų sąryšį. Laiko logikose yra naudojami šie logikos operatoriai: $\circ, \diamond, \square, U, A, E$. Jų apibrėžimai nurodomi tolimesniuose skyriuose.

Verta paminėti, jog skirtingos laiko logikos naudoja skirtingus laiko operatorius. Tiesinėje laiko logikoje paprastai naudojami $\square F$ ir $\circ F$ arba kaip alternatyva šiems dviems operatoriams vienas operatorius FUG (atsisakoma A ir E žymėjimų, kadangi yra tik vienas galimas kelias), o skaidaus laiko logikoje naudojami sudėtingesni $A(FUG)$ ir $E(FUG)$ operatoriai.

3.3.1. Tiesinė laiko logika

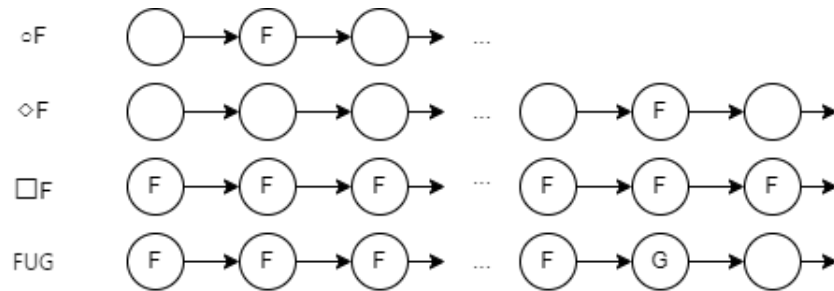
Tiesinė laiko logika aprašo įvykius, kurie vyksta vientisame laike, t.y. nėra jokių laiko išsišakojimų ir įmanoma tik viena laiko linija. Tiesinė laiko logika yra atskiras atvejis sudėtingesnės, skaidaus laiko logikos, kurioje įmanomas laiko išsišakojimas [Arz21].

Formulės teisingumo apibrėžimas toks pats kaip klasikinės modalumo logikos formulės teisingumo apibrėžimas, tik papildomas modalumais.

Toliau apibrėžiama tiesinės laiko logikos operatorių semantika [Bir10]:

1. $\circ F$ - F teisinga kitoje (*angl.* next) kelio viršūnėje
2. $\diamond F$ - F teisinga kurioje nors kelio viršūnėje (egzistuoja ateities laiko momentas t' , kuriame F teisinga)
3. $\square F$ - F teisinga visose kelio viršūnėse
4. FUG - F yra teisinga iki tol kol G tampa teisinga

Tiesinės laiko logikos operatorių semantiką aiškinančios schemas - Kripke pasauliai (Pav. 2) [Nor04]. F raide pažymėtuose grafinės interpretacijos pasauliuose, formulė F yra teisinga, tuščiuose pasauliuose formulės F teisingumas, operatoriaus išaiškinimo prasme, neturi reikšmės.



2 pav. Tiesinės laiko logikos operatorių semantikos grafinė interpretacija

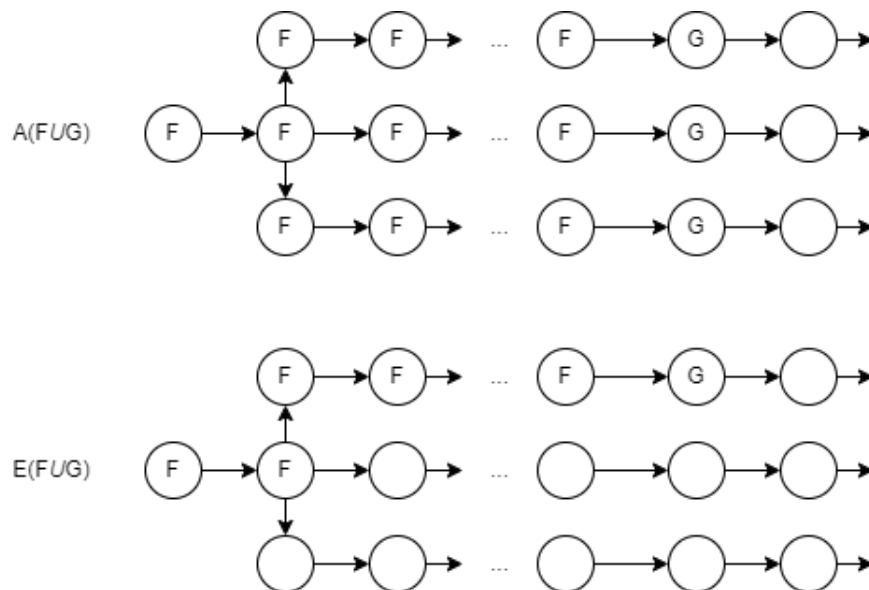
3.3.2. Skaidaus laiko logika

Skaidaus laiko logikoje galimi du ar daugiau išsišakojimų laike. Kitaip tariant, tam tikrame laiko momente, laiko liniją galime išskaidyti į kelias.

Skaidaus laiko logikos operatorių semantiką aiškinančios schemas - Kripke pasauliai (Pav. 3). F ir G raidėmis pažymėtuose grafinės interpretacijos pasauliuose, formulės F, G yra teisingos, tuščiuose pasauliuose formulių F, G teisingumas, operatoriaus išaiškinimo prasme, neturi reikšmės.

Toliau apibrėžiama skaidaus laiko logikos operatorių semantika [Bir10]:

1. $A(FUG)$ - kiekvienam laiko keliui $t_1, t_2, \dots$ egzistuoja laiko momentas t , kad F yra teisingas kiekvienam laiko momente $t', t_1 \leq t' \leq t$ ir G yra teisingas laiko momentu t .
2. $E(FUG)$ - egzistuoja laiko kelias $t_1, t_2, \dots$ ir laiko momentas t , kad F yra teisingas kiekvienam laiko momente $t', t_1 \leq t' \leq t$ ir G yra teisingas laiko momentu t .



3 pav. Skaidaus laiko logikos operatorių semantikos grafinė interpretacija

3.3.3. Žinomi sekvenciniai skaičiavimai tiesinei laiko logikai

Tiesinei laiko logikai yra sugalvotas ne vienas sekvencinio skaičiavimo variantas. Šiame skyrelyje aptarsime tris pagrindinius [AP13].

18 Apibrėžimas. *Begalinis tiesinės laiko logikos sekvencinis skaičiavimas yra skaičiavimas su aksioma $\Gamma, F \vdash \Delta, F$, loginėmis taisyklėmis $(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg)$ ir modalinėmis taisyklėmis:*

$$\begin{aligned}
 (\circ) \quad & \frac{\Gamma \vdash \Delta}{\Pi, \circ\Gamma \vdash \Theta, \circ\Delta} & (\Box \vdash) \quad & \frac{A, \circ\Box A, \Gamma \vdash \Delta}{\Box A, \Gamma \vdash \Delta} \\
 (\vdash \Box_w) \quad & \frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, \circ A \quad \dots \quad \Gamma \vdash \Delta, \overbrace{\circ \dots \circ}^n A}{\Gamma \vdash \Delta, \Box A}
 \end{aligned}$$

Čia A ir B yra formulės, $\Gamma, \Delta, \Pi, \Theta$ žymi baigtines, galimai tuščias formulių aibes. $\circ\Gamma, \circ\Delta$ žymi aibes formulių, suvaržytų išoriniu $\circ$ operatoriumi.

Šio skaičiavimo pagrindinė problema yra begalinis sekvencijų kiekis, atsiradęs pritaikant $(\vdash \Box_w)$ taisyklę. Informatikos moksle sunku (arba neįmanoma) automatizuoti proceso, kuriame vyrauja begalinės sekos. Taikant šį sekvencinį skaičiavimą nesinaudojant automatizavimo priemonėmis, nesudėtinga pritaikius $(\vdash \Box_w)$ taisyklę, nerašyti begalinio kiekio sekvencijų, paprasčiausiai pasižymėti abstraktų skaičių n ir išvedimo eigoje jį inicializuoti į tinkamą, kad išvedimas būtų sėkmingas. Iš gerosios pusės, šis skaičiavimas nereikalauja naujos formulės konstravimo, kuris apsunkina ir automatizuotą ir neautomatizuotą išvedimą.

19 Apibrėžimas. *Invariantinės taisyklės tiesinės laiko logikos sekvencinis skaičiavimas yra skaičiavimas gaunamas iš begalinio tiesinės laiko logikos sekvencinio skaičiavimo taisyklių, tik pakeitus $(\vdash \Box_w)$ taisyklę į:*

$$(\vdash \Box_I) \quad \frac{\Gamma \vdash \Delta, I \quad \Gamma \vdash \Delta, \circ I \quad I \vdash A}{\Gamma \vdash \Delta, \Box A}$$

Čia I (vadinama invariantine formule) yra nauja formulė, dažnai konstruojama iš sekvencijos $\Gamma \vdash \Delta, \Box A$ formulių.

Invariantinės taisyklės skaičiavimas taip pat sukelia bėdų, norint automatizuoti sekvencijos išvedimą, kadangi taisyklėje $(\vdash \Box_I)$ atsiranda nauja formulė, kurią reikia sugalvoti tinkamai, kad tolimesnis išvedimas būtų sėkmingas. Tinkamos invariantinės formulės sukonstravimas gali būti ne toks sudėtingas uždavinys kai numatoma tolimesnė išvedimo eiga ir neperrenkami visi įmanomi formulės variantai. Automatizuoti šį procesą būtų labai sunku, nes neaišku kaip sugalvoti tinkamą formulę pritaikius $(\vdash \Box_I)$ taisyklę. Literatūroje taip pat nagrinėjami šio skaičiavimo variantai su apribota invariantine formule ar konstruktyvūs metodai invariantinei formulei rasti [Val80].

20 Apibrėžimas. *Primarinė sekvencija yra sekvencija pavidalo $\Sigma, \circ\Gamma \vdash \circ\Delta, \Pi$, kur Σ, Π - tik atominių formulių aibės.*

21 Apibrėžimas. *Taisyklės $(\circ_p)$ apibrėžimas.*

$$(\circ_p) \frac{\Gamma \vdash \Delta}{\Sigma, \circ\Gamma \vdash \circ\Delta, \Pi}$$

Šios taisyklės išvada yra primarinė sekvencija.

22 Apibrėžimas. *Neigiamas ciklas $S \rightsquigarrow S'$ yra išvedimo paieškos medžio fragmentas, kur S' yra virš S ir $S' \leq S$. Čia S' antecedentas yra S antecedento poaibis ir S' sukcedentas yra S sekvencijos sukcedento poaibis.*

Pavyzdys. Toliau pateikiamas neigiamo ciklo atsiradimo pavyzdys, bandant išvesti sekvenciją $F, \circ F, F \vee G \vdash H, \circ G, G$ begaliniu tiesinės laiko logikos sekvenciniu skaičiavimu.

$$(\vee \vdash) \frac{(\circ) \frac{S' : F \vdash G}{F, F, \circ F \vdash H, \circ G, G} \quad (\circ) \frac{S'' : F \vdash G}{G, F, \circ F \vdash H, \circ G, G}}{S : F, \circ F, F \vee G \vdash H, \circ G, G}$$

Pastebėkime, jog gavome sekvencijas S' ir S'' , kurios yra aukščiau paieškos išvedimo medyje nei pradinė sekvencija S ir tenkinamos sąlygos $S' \leq S, S'' \leq S$. Taigi gavome du neigiamus ciklus $S \rightsquigarrow S'$ ir $S \rightsquigarrow S''$. Ši sekvencija yra neišvedama. Konstruojant išvedimo paieškos medį ir gavus neigiamą ciklą dažniausiai skaičiavimuose tolimesnis šios šakos konstravimas yra beprasmis. Jeigu išvedama S' , tai galima rasti ir analogišką sekvencijos S išvedimo medį. Taigi bet kokiai išvedamai sekvencijai galima rasti išvedimo medį be neigiamų ciklų, o tai reiškia, kad paieškoje galima apsiriboti tik išvedimo paieškos medžiais be neigiamų ciklų. Tokių ciklų aptikimas automatinėmis priemonėmis yra brangi operacija.

23 Apibrėžimas. *Teigiamas ciklas $S \rightsquigarrow S'$ yra išvedimo paieškos medžio fragmentas, kur S' yra virš S ir $S' \geq S$. Čia yra S' sudaryta iš tų pačių ir galimai papildomų formulių.*

24 Apibrėžimas. *Ciklu $S \rightsquigarrow S'$ vadinsime teigiamą arba neigiamą ciklą.*

25 Apibrėžimas. *Indukcinė laiko operatoriaus taisyklė $(\vdash \Box_L)$ yra:*

$$(\vdash \Box_L) \frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, \circ\Box A}{\Gamma \vdash \Delta, \Box A}$$

26 Apibrėžimas. *Sekvencija S' yra ciklinio tipo aksioma, tada ir tik tai tada, kai:*

1. $S \rightsquigarrow S'$ yra teigiamas ciklas
2. S' pasirodo dešinėje išvedimo šakoje S atžvilgiu pritaikius $(\vdash \Box_L)$ taisyklę ir $S \rightsquigarrow S'$ nėra nei vieno taisyklės $(\vdash \Box_L)$ taikymo su kairiąja šaka.

27 Apibrėžimas. *Ciklinio tipo laiko logikos sekvencinis skaičiavimas LTL yra skaičiavimas su taisyklėmis $(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\Box \vdash), (\vdash \Box_L), (\circ)$ ir ciklinio tipo aksioma, bei aksioma $\Gamma, F \vdash \Delta, F$.*

Pastaba. Toliau šiame darbe ciklinio tipo laiko logikos sekvencinį skaičiavimą vadinsime klasikiniu sekvenciniu skaičiavimu tiesinėje laiko logikoje arba *LTL*.

Išvedimo eigoje atsiradęs neigiamas ciklas, dažniausiai lemia, jog sekvencija S' bus neišvedama. Literatūroje toks ciklas vadinamas "*bloguoju*" ciklu. Jeigu išvedime gauname tikslių ciklą $S = S'$ ir tai nėra ciklinė aksioma, tai reiškia, jog ši šaka neišvedama, nes jos išvedimas bus begalinis. Tačiau tai nereiškia, jog neišvedama sekvencija S . Sekvencijai S galima sukonstruoti k skirtingų išvedimo paieškos medžių, iš kurių keli turės neigiamus arba tikslius ciklus ir dėl to jie nebus išvedimo medžiais (visi medžio lapai nesibaigs aksiomomis). Tuo pačiu galbūt kažkuris iš kitų S išvedimo paieškos medžių neturi ciklų ir baigiasi aksiomomis, o tai reikštų, kad sekvencija S visgi išvedama. Dažnai keliamas tikslas yra aptikti tokius ciklus, norint gauti baigtinį skaičiavimą. Kai kuriuose skaičiavimuose tokie ciklai negalimi, o kai kuriuose jie gali atsirasti. Ciklinės aksiomos literatūroje dar vadinamos "*geraisiais*" ciklais.

Pirmasis ciklinio tipo sekvencinį skaičiavimą aprašė P. Wolper 1985 metais [Wol85]. Autorius pastebėjo, jog ne klasikinei logikai nepakanka paprastosios $F \vdash F$ aksiomos ir norint išvedimo medį laikyti įrodymu reikia papildomų suvaržymų (šiuo atveju ciklų), kuriuos reikia aptikti. Laikui bėgant efektyvus ciklų aptikimas ar fragmentų konstravimas norint išvengti neefektyvaus ciklų tikrinimo pritaikytas BDI logikai [Bir10], DPDL logikai [Pli08] ir daugeliui kitų ne klasikinių logikų.

Ciklinio tipo sekvencinio skaičiavimo pilnumas ir išsprendžiamumas įrodytas [APP⁺20] 2020 metais. Šį sekvencinį skaičiavimą galima nesudėtingai automatizuoti, tačiau ciklinių aksiomų tikrinimas naudoja daug resursų ir yra labai brangi operacija. Natūraliai kyla poreikis mokėti nustatyti galimą ciklą iš anksto netikrinant ankščiau išvedimo medyje esančių sekvencijų. Ciklų egzistavimo galimybė priklauso nuo sekvencijoje naudojamų formulių struktūros, ypač nuo $\Box$ ir $\circ$ operatorių kiekio ir pozicijos formulėse.

Pavyzdys. Toliau pateikiamas ciklinės aksiomos atsiradimo pavyzdys skaičiavime *LTL*, išvedinėjant sekvenciją $S : \Box P \vdash \Box(P \vee Q)$. Pastebime, jog, dešinėje medžio išvedimo šakoje S atžvilgiu, pritaikę taisyklę $(\vdash \Box_L)$ gauname tokią pačią sekvenciją S' . Tai ir yra vadinama ciklu, kadangi išvedinėjant sekvenciją S' vėl gausime tokią pačią sekvenciją dešinėje medžio išvedimo šakoje S' atžvilgiu. Šiuo atveju gavome ciklinę aksiomą, kadangi įgyvendintos abi ciklinės aksiomos sąlygos: $S \rightsquigarrow S'$ yra teigiamas ciklas ir S' pasirodo dešinėje išvedimo šakoje S atžvilgiu pritaikius $(\vdash \Box_L)$ taisyklę.

$$\begin{array}{c}
 (\vdash \vee) \quad \frac{P, \circ \Box P \vdash P, Q}{P, \circ \Box P \vdash P \vee Q} \qquad (\circ) \quad \frac{S' : \Box P \vdash \Box(P \vee Q)}{P, \circ \Box P \vdash \circ \Box(P \vee Q)} \\
 (\vdash \Box_L) \quad \frac{\qquad}{\qquad} \qquad (\Box \vdash) \quad \frac{P, \circ \Box P \vdash \Box(P \vee Q)}{S : \Box P \vdash \Box(P \vee Q)}
 \end{array}$$

3 Teiginys. *Įrodyta, jog aprašyti sekvenciniai skaičiavimai (begalinis, invariantinės taisyklės ir ciklinio tipo) tiesinei laiko logikai yra ekvivalentūs ir pilni. [AP13].*

4. Laiko logikos pritaikymo sferos

Laiko logika turi daugybę pritaikymo sferų informatikos srityje. Tarp jų yra ir tradicinių, gerai išvystytų sričių, tokių kaip programų specifikavimas ir verifikavimas [MP91; Pnu06], laikinosios duomenų bazės [Cho94], paskirstytos ir multi-agentų sistemos [FHM⁺95; TD16] ir naujesnių - žinių reprezentacija ir samprotavimas [AF94; Sch93].

4.1. Laikinosios duomenų bazės

Laiko logika gali būti pritaikyta duomenų bazėms ir užklausoms, kuriomis galime iš duomenų bazės išgauti informaciją. Laiko duomenų bazė yra saugykla, kurioje saugoma informacija turinti prasmę laiko atžvilgiu. Laiko užklausų kalba yra užklausų kalba skirta laiko duomenų bazėms. Pasitelkdami laiko logiką, galime specifiškai užklausti, kuriose nurodomi su laiku susiję parametrai. Pavyzdžiui, užklausa "pateikite sąrašą šalių, kurios prarado ir atgavo nepriklausomybę", galėtų būti formuojama su operatorių *until* ir *since* pagalba. J. Chomski pateiktoje apžvalgoje apie laiko užklausos kalbas [Cho94] formaliai apibrėžiamas laiko domenai ir laiko duomenų bazė ir naudojantis šiais apibrėžimais apžvelgiamos laiko užklausos kalbos. Šioje apžvalgoje pasiūlyta formali sistema, skirta laiko duomenų bazėms ir laiko užklausų kalboms tirti.

4.2. Žinių reprezentacija ir samprotavimas

Tam tikriems samprotavimams ir žinių reprezentacijai taip pat gali būti pasitelkta laiko logika. A. Artale ir E. Franconi savo darbe [AF94] tyrė formalią sistemą, leidžiančią nagrinėti laiką, veiksmus ir planus tuo pačiu vienu būdu. Autoriai pristatė formalią kalbą, skirtą laikui ir veiksmams išreikšti ir apie juos samprotauti. Jų darbe veiksmas išreiškiamas apibrėžiant įvykių eigą laike veiksmo vykymo metu. Veiksmas apibrėžiamas per laiko apribojimus pasaulio būsenoms (kurios priklauso pačiam veiksmui) ir paprastesniems veiksmams vykstantiems per tam tikrą laiką.

4.3. Multi-agentų sistemos

Pastaraisiais metais daug dėmesio buvo skiriama multi-agentų sistemoms. Šios sistemos gali būti sudarytos iš tam tikrų regionų ($A, B, C, \dots$) ir robotų, kurie yra atsakingi už juos. Tam tikrais atvejais gali susidaryti tokios sąlygos, kurioms galiojant vienas agentas gali būti nepajėgus įgyvendinti jam priskirtų tikslų, todėl turi bendradarbiauti su kitais agentais. J. Tumova ir D. Dimarogonas savo darbe tyrė multi-agentų sistemas, kur kiekvienas agentas yra modeliuojamas kaip diskreti būsenų perėjimų sistema aprašyta tiesine laiko logikos formule. Tradiciniai baigtiniais automatais grindžiami metodai, skirti multi-agentams valdyti remiasi pilna agentų komandos sinchronizacija po kiekvieno agentų diskretaus žingsnio, todėl reikalauja didelių skaičiavimo sąnaudų [TD16]. Vietoje agentų specifikacijos vertinimo kaip tiesinės laiko logikos formulių konjunkcijos per visą agentų komandą, autoriai pasiūlė būdą nagrinėti tiesinės laiko logikos formulės įgyvendinamumą iš lokalios perspektyvos.

4.4. Veiksmų laiko logika

Veiksmų laiko logika (*angl.* Temporal Logic of Actions - TLA) yra logika, leidžianti nagrinėti ir specifikuoti lygiagrečiai vykdomas sistemas [Lam93] aprašyta L.Lamport.

Lygiagretūs algoritmai dažniausiai yra specifikuojami kokia nors programa. Algoritmo korektiškumas nusakomas tuo, jog programa patenkina tam tikrą savybę. Veiksmų laiko logikos pagrindinė idėja yra algoritmo ir norimos patenkinti savybės nusakymas vienos (TLA) logikos formulėmis. Motyvacija tokiam sprendimui kyla ne iš noro nusakyti tam tikras sistemas elegantiškiau, o iš praktinės problemos, kuri yra samprotavimas apie realius algoritmus. Vienintelis būdas išvengti subtilių problemų, kylančių programuojant lygiagrečius algoritmus, yra griežtas ir konkretus argumentavimas. Iš čia kyla motyvacija padaryti šį argumentavimą kuo paprastesniu, kad programos formalizavimas irgi būtų paprastas ir intuityvus [Lam93].

5. Fragmentai

Egzistuojant tam tikroms problemoms logikoje, pavyzdžiui, kai nežinomas išsprendžiamumas, baigtinumas ar egzistuoja didelis sudėtingumas, nagrinėjami logikos fragmentai ir bandoma įrodyti, jog apsibrėžtuose fragmentuose minėtų problemų nėra. Kitaip tariant specifiniams fragmentams įrodomas išsprendžiamumas, baigtinumas ar pateikiamas efektyvesnis skaičiavimas.

28 Apibrėžimas. *Tam tikros logikos **fragmentas** yra sintaksiškai suvaržytas visos logikos formulių poaibis [AZ07]. Išskirto fragmento ir visos logikos semantika sutampa, taigi bet kokia fragmente apibrėžta formulė, gali būti išreikšta originalioje logikoje.*

29 Apibrėžimas. *Unarinis fragmentas yra fragmentas, kuriame formulė gali būti suvaržyta ne daugiau nei vienu būtinumo operatoriumi.*

5.1. Naujo fragmento apibrėžimas

Šiame skyriuje aprašomas fragmentas, kuriam bus formuojamas naujas, efektyvesnis tiesinės laiko logikos sekvencinis skaičiavimas.

30 Apibrėžimas. *Apibrėšime modalumo logikos fragmentą α_1 , kuriame:*

1. *Loginis kintamasis yra formulė.*
2. *Jei F yra be-modalinė formulė, tai $\Box F$ irgi formulė*
3. *Jei F formulė, tai $\neg F$ irgi formulė*
4. *Jei F, G formulės, tai $F \vee G$ ir $F \wedge G$ taip pat formulės.*

31 Apibrėžimas. *Apibrėšime išplėstinį α_1 modalumo logikos fragmentą α_* , kuriame:*

1. *Loginis kintamasis yra formulė.*
2. *Jei F yra be-modalinė formulė, tai $\Box F$ irgi formulė*
3. *Jei F formulė, tai $\neg F$ irgi formulė*
4. *Jei $\Box F$ formulė, tai $\circ\Box F$ irgi formulė*
5. *Jei F, G formulės, tai $F \vee G$ ir $F \wedge G$ taip pat formulės.*

Pastaba. Sekvencijos išvedime $LTL1$ fragmentui α_1 gausime formules ir iš išplėstinio α_* fragmento. Dažniausiai sekvenciniuose skaičiavimuose turint sekvenciją sudarytą iš fragmento formulių visose sekvencijose, esančiose išvedimo medyje, visos formulės bus pradinės sekvencijos formulių poformuliai, todėl irgi priklausys tam pačiam fragmentui. Laiko logikai skirtuose sekvenciniuose skaičiavimuose paieškos išvedimo medyje galima gauti ir formules, kurios nėra pradinės sekvencijos formulių poformulis, todėl apibrėžiame išplėstinį fragmentą. Tai reiškia, kad tikrinama sekvencija privalo priklausyti α_1 fragmentui, o išvedimo medyje sutinkamos sekvencijos formulės gali priklausyti ir išplėstiniam fragmentui α_* .

Fragmentas α_1 yra unarinis tiesinės laiko logikos fragmentas. Šio fragmento formulių pavyzdžiai galėtų būti:

- $\Box G$
- $F \vee \Box G$
- $\Box(G \wedge \neg F)$

Formulės nepatenkančios į fragmentą:

- $\Box P \vee \Box(Q \wedge \Box P)$
- $\Box\Box P \vee \Box Q$
- $\circ\Box(\circ P \wedge \Box Q) \vee \Box Q$

Šiose pavyzdinėse formulėse, nepatenkančiose į išskirtą fragmentą, išorinis būtinumo operatorius įtakoja vidinių formulių veikimo sritį taip pažeisdamos fragmento apibrėžimo 2. punktą.

2 Lema. Jeigu *LTL* skaičiavime išvedama sekvencija $S = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n$ iš fragmento α_1 , kur $\forall i, F_i$ - be-modulinė formulė, tada išvedama

1. $\Pi, \Box\Gamma \vdash \Sigma$, arba
2. $\Box\Gamma \vdash \Box F_1, \dots, \Box F_n$

Irodymas. Sekvencijai S taikome visas galimas ($\vdash \Box_L$) taisykles ir gauname 2^n sekvencijų, kurios visos yra išvedamos, kadangi buvo taikytos tik apverčiamos taisyklės:

- $\Pi, \Box\Gamma \vdash \Sigma, F_1, \dots, F_n$
- $\Pi, \Box\Gamma \vdash \Sigma, F_1, \dots, \circ\Box F_n$
- ...
- $\Pi, \Box\Gamma \vdash \Sigma, \circ\Box F_1, \dots, \circ\Box F_n$

Viena iš šių sekvencijų yra ir $S_1 = \Pi, \Box\Gamma \vdash \Sigma, \circ\Box F_1, \dots, \circ\Box F_n$ (gauta visada imant tik dešinę taisyklės ($\vdash \Box_L$) taikymo šaką). Sekvencijai S_1 taikome visas galimas ($\Box \vdash$) taisykles ir gauname sekvenciją $S_2 = \Pi, \Gamma, \circ\Box\Gamma \vdash \Sigma, \circ\Box F_1, \dots, \circ\Box F_n$. Π, Γ, Σ yra be-modulinės formulių aibės, nes sekvencija S yra iš fragmento α_1 , taip pat šiose aibėse nebus formulių suvaržytų $\circ$ operatoriumi. Taikytos taisyklės ($\Box \vdash$) yra apverčiamos ir S_1 išvedama, tai S_2 irgi išvedama. Kadangi S_2 išvedama, tai:

- a) išvedama $\Pi, \Gamma \vdash \Sigma$, arba
- b) išvedama $\circ\Box\Gamma \vdash \circ\Box F_1, \dots, \circ\Box F_n$

Atveju (a), bus išvedama ir $\Pi, \Box\Gamma \vdash \Sigma$, nes jai pritaikius visas ($\Box \vdash$) gausime sekvenciją $\Pi, \Gamma, \circ\Box\Gamma \vdash \Sigma$, kuri išvedama, nes išvedama $\Pi, \Gamma \vdash \Sigma$.

Atveju (b), bus išvedama ir $\Box\Gamma \vdash \Box F_1, \dots, \Box F_n$, nes išvedama $\circ\Box\Gamma \vdash \circ\Box F_1, \dots, \circ\Box F_n$ ir jai galima pritaikyti tik taisyklę ($\circ$).

Pastaba. Kadangi sekvencija S yra iš fragmento α_1 visos formulės F_i yra be-modulinės formulės.

3 Lema. Jeigu *LTL* skaičiavime išvedama sekvencija $S = \Box\Gamma \vdash \Box F_1, \dots, \Box F_n$ iš fragmento α_1 ir $\Box F_i$ yra bet kuri modalinė formulė iš S sukcedento, tai galimi 2 atvejai:

1. Išvedama sekvencija $\Box\Gamma \vdash \Box F_1, \dots, \Box F_{i-1}, \Box F_{i+1}, \dots, \Box F_n$, arba
2. Išvedama sekvencija $\Gamma \vdash F_i$

Čia $\forall i, F_i$ - be-modulinė formulė.

Irodymas. Pradinei sekvencijai S taikome visas įmanomas ($\Box \vdash$) taisykles ir taip gauname sekvenciją $S_2 = \Gamma, \circ\Box\Gamma \vdash \Box F_1, \dots, \Box F_n$. S_2 išvedama, kadangi išvedama S ir taikytos tik apverčiamos taisyklės.

Sekvencijai S_2 taikome visas įmanomas ($\vdash \Box_L$) taisykles ir gauname 2^n sekvencijų, kurios visos yra išvedamos, kadangi taikytos tik apverčiamos taisyklės:

- $\Gamma, \circ\Box\Gamma \vdash F_1, \dots, F_n$
- $\Gamma, \circ\Box\Gamma \vdash F_1, \dots, \circ\Box F_n$
- ...
- $\Gamma, \circ\Box\Gamma \vdash \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$
- ...
- $\Gamma, \circ\Box\Gamma \vdash \circ\Box F_1, \dots, \circ\Box F_n$

Viena iš šių sekvencijų yra ir $S_3 = \Gamma, \circ\Box\Gamma \vdash \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$, kurios sukcedente visos formulės F yra su $\circ\Box$ operatoriais, išskyrus F_i .

Dabar sekvencijai S_3 pritaikome visas logines taisykles ($\vee, \wedge, \neg$) formulėms iš Γ ir gauname m naujų sekvencijų, kurių antecedente ir sukcedente galimai atsirado papildomų atominių formulių Γ_i, Γ'_i . Jeigu išvedimo eigoje gauname aksiomą, toliau šiai šakai taikome logines taisykles tol, kol tą galima daryti. Visose šiose sekvencijose formulės suvaržytos su $\circ$ yra tokios pačios, kadangi Γ sudaryta tik iš be-modulinių formulių, nes sekvencija iš fragmento α_1 :

- $\Gamma_1, \circ\Box\Gamma \vdash \Gamma'_1, \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$
- $\Gamma_2, \circ\Box\Gamma \vdash \Gamma'_2, \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$
- ...
- $\Gamma_z, \circ\Box\Gamma \vdash \Gamma'_z, \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$
- ...
- $\Gamma_m, \circ\Box\Gamma \vdash \Gamma'_m, \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$

Tegul $S_4 = \Gamma_z, \circ\Box\Gamma \vdash \Gamma'_z, \circ\Box F_1, \dots, \circ\Box F_{i-1}, F_i, \circ\Box F_{i+1} \dots, \circ\Box F_n$ viena iš šių sekvencijų. Ši sekvencija yra išvedama, kadangi ji gauta iš S_3 taikant tik apverčiamas taisykles. Taip pat Γ_z ir Γ'_z sudarytos tik iš atominių formulių, ir jose nėra modalinių operatorių.

Turime du galimus atvejus:

1. Sekvencija S_4 išvedama taikant taisyklę $(\circ)$
2. Sekvencija S_4 išvedama netaikant taisyklės $(\circ)$

Taip yra todėl, kad sekvencijai S_4 galime taikyti logines taisykles formulei F_i (2 atvejis), arba taikyti taisyklę $(\circ)$ (1 atvejis) ir ji išvedama. Todėl viena iš šių sekvencijų yra išvedama:

- $S_5 = \Gamma_z \vdash \Gamma'_z, F_i$, arba
- $S_6 = \circ\Box\Gamma \vdash \circ\Box F_{i-1}, \circ\Box F_{i+1} \dots, \circ\Box F_n$

Iš to seka, kad:

- S_5 išvedama (pati yra aksioma arba gauname aksiomas šios sekvencijos išvedimo medžio lapuose, taikant logines taisykles F_i), arba
- Išvedama sekvencija S_6 , kuriai taikoma taisyklė $(\circ)$.

Galiausiai gauname du atvejus:

1. $\exists z \in \{1, 2, \dots, m\}$, kad išvedama S_6 .
Tada išvedama ir $S_1 = \Box\Gamma \vdash \Box F_1, \dots, \Box F_{i-1}, \Box F_{i+1}, \dots, \Box F_n$.
2. $\forall z \in \{1, 2, \dots, m\}$ išvedamos $\Gamma_z \vdash \Gamma'_z, F_i$. Tada išvedama ir $\Gamma \vdash F_i$

Pastaba. Kadangi sekvencija S yra iš fragmento α_1 visos formulės F_i yra be-modulinės formulės.

4 Lema. Jeigu sekvencija $S = \Box\Gamma \vdash \Box F_1, \dots, \Box F_n$ iš fragmento α_1 išvedama skaičiavime LTL , tai egzistuoja toks i , kad išvedama ir sekvencija $S'_1 = \Box\Gamma \vdash F_i$ arba išvedama $S''_1 = \Box\Gamma \vdash$.

Irodymas. Iš 3 lemos žinome, jog jeigu išvedama S , tai

1. Išvedama sekvencija $\Box\Gamma \vdash \Box F_1, \dots, \Box F_{i-1}, \Box F_{i+1}, \dots, \Box F_n$, arba
2. Išvedama sekvencija $\Gamma \vdash F_i$

Sakykime egzistuoja toks i , kad išvedama $S_2 = \Gamma \vdash F_i$ (3 lemos 2 atvejis). Tada parodysime, kad išvedama S'_1 .

Sekvencijai S'_1 pritaikome taisyklę $(\Box \vdash)$ visoms formulėms iš $\Box\Gamma$ ir silpninimą. Pasižymėkime l formulių kiekį aibėje Γ , o atsiradusių naujų formulių aibę $\circ\Box\Gamma'$.

$$\begin{array}{c} (Weak) \frac{\Gamma \vdash F_i}{\circ\Box\Gamma', \Gamma \vdash F_i} \\ (\Box \vdash) * l \frac{\vdots}{S'_1 = \Box\Gamma \vdash F_i} \end{array}$$

Pritaikius taisykles gavome $\Gamma \vdash F_i$ (kuri yra išvedama), vadinasi išvedama ir S'_1 .

Jei toks i neegzistuoja, tai parodysime, kad išvedama ir S''_1 . Šiuo atveju galioja 3 lemos 1 atvejis. Su kiekvienu i , $\Box F_i$ yra nebūtina ir gali būti pašalinta iš išvedimo medžio su silpninimo taisykle. Pašalinus visas $\Box F_i$ lieka išvedama sekvencija S''_1 .

Pastaba. Kitaip tariant, jeigu turime sekvenciją $S = \Box\Gamma \vdash \Box F_1, \dots, \Box F_n$ iš fragmento α_1 įrodėme, jog:

- jeigu išvedama $\Box\Gamma \vdash F_i$, tai reiškia, jog pakanka tik vienos, kurios nors sukcedento formulės F_i norint išvesti sekvenciją S , arba
- jeigu išvedama $\Box\Gamma \vdash$, tai reiškia, kad jokia F_i nedalyvauja tolimesniame išvedime.

7. Skaičiavimas $LTL1$

Šiame skyriuje apibrėžiamas naujas sekvencinis skaičiavimas $LTL1$ fragmentui α_1 . Pateikiami sekvencijų išvedimo pavyzdžiai $LTL1$ skaičiavime, kurie lyginami su ciklinio tipo tiesinės laiko logikos sekvenciniu skaičiavimu. Įrodomas minėtų skaičiavimų ekvivalentumas sekvencijoms iš fragmento α_1 . Galiausiai pateikiamas skaičiavimų LTL ir $LTL1$ palyginimas.

7.1. Sekvencinio skaičiavimo $LTL1$ apibrėžimas

Šiame skyriuje pristatomas sekvencinis skaičiavimas tiesinėje laiko logikoje fragmentui α_1 .

32 Apibrėžimas. *Taisyklės $(\vdash \Box_{L*})$ apibrėžimas.*

$$(\vdash \Box_{L^*}) \quad \frac{\Box\Gamma \vdash B_1 \quad \parallel \quad \dots \quad \parallel \quad \Box\Gamma \vdash B_m \quad \parallel \quad \Pi, \Box\Gamma \vdash \Sigma}{\Pi, \Box\Gamma \vdash \Sigma, \Box B_1, \dots, \Box B_m},$$

čia $n, m \in \mathbb{N}$, $\square\Gamma = \square A_1, \dots, \square A_n$. Π, Σ - atominės, be-modulinės formulės.

33 Apibrėžimas. *LTL1 yra tiesinės laiko logikos sekvencinis skaičiavimas su taisyklėmis $(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\Box \vdash), (\vdash \Box_L^*), (\odot)$ ir aksioma $\Gamma, F \vdash \Delta, F$.*

$LTL1$ yra tiesinės laiko logikos sekvencinis skaičiavimas gaunamas iš ciklinio tipo laiko logikos sekvencinio skaičiavimo, panaikinus ciklinę aksiomą ir taisyklę ($Weak$) ir pakeitus taisyklę $(\vdash \Box_L)$ nauja taisykle $(\vdash \Box_{L*})$.

Šiame sekvenciniame skaičiavime eliminuojamas ciklinių aksiomų naudojimas ir ciklą, kuriuos lemia taisyklės ($\vdash \Box_L$) taikymas, atsiradimas. Ši savybė remiasi pastebėjimu, jog pritaikius ($\vdash \Box_L$) taisyklę tam tikro tipo sekvencijai, po taisyklės pritaikymo, dešiniausioje medžio šakoje visuomet atsiras ciklinė aksioma. Kitaip tariant, $\Pi, \Box\Gamma \vdash \Sigma, \Box B_1, \dots, \Box B_m$ tipo sekvencija lemia ciklą atsiradimą.

Likusiam medyje lieka įrodyti sekvenčijas, kurių antecedentas yra vienodas. Jis susidaro iš *visų* formulių esančių pradinės sekvenčijos antecedente, kurios yra apribotos būtinumo operatoriaus. Naujai atsiradusių sekvenčių sukcedentą sudaro *atskirai* paimtos formulės iš pradinės sekvenčijos sukcedento, kurios taip pat yra apribotos būtinumo operatoriaus, tačiau užrašomos be $\Box$ operatoriaus. Šį pastebėjimą iliustruosime pavyzdžiu pateikdami išvedimo medžio fragmentą sekvenčiai $\Box F \vdash \Box H, \Box G$ skaičiavime *LTL*:

$$\begin{array}{c}
loop \oplus \\
\frac{}{\Box F \vdash \Box H} \\
\frac{\vdots}{\Box F \vdash H} \quad \frac{\frac{}{\Box F \vdash \Box H}}{\circ \Box F, F \vdash \circ \Box H} \\
\frac{}{\Box F \vdash H} \quad \frac{}{\Box F \vdash \circ \Box H} \\
\frac{}{\Box F \vdash \Box H} \\
\frac{\vdots}{\Box F \vdash H, G} \quad \frac{\frac{}{\Box F \vdash \Box H}}{\circ \Box F, F \vdash \circ \Box H, G} \\
\frac{}{\Box F \vdash H, G} \quad \frac{}{\Box F \vdash \circ \Box H, G} \\
\frac{}{\Box F \vdash \Box H, G} \quad S_1 \\
\frac{}{\Box F \vdash \Box H, \circ \Box G} \\
\frac{}{\Box F \vdash \Box H, \Box G}
\end{array}$$

Sekvencijos $\Box(p \wedge q) \vdash \Box p, \Box q$ išvedimas skaičiavime $LTL1$ atrodo taip:

$$\frac{\frac{\text{neišvedama}}{\vdots} \quad \frac{\frac{\oplus}{\circ\Box(p \wedge q), p, q \vdash q}}{\circ\Box(p \wedge q), p \wedge q \vdash q}}{\Box(p \wedge q) \vdash p \wedge x \quad || \quad \Box(p \wedge q) \vdash q} \quad \Box(p \wedge q) \vdash \Box(p \wedge x), \Box q$$

Pastebime, jog skaičiavimas labai supaprastėja - jame nebelieka ciklinių aksiomų ir išvedimo medis tampa daug mažesnis. Pastebėkime, jog naujos taisyklės pagalba užtenka įrodyti vieną išvedamą šaką. Pavyzdyje šaka $\Box(p \wedge q) \vdash p \wedge x$ neišvedama, tačiau išvedama $\Box(p \wedge q) \vdash q$, o to pakanka norint įrodyti sekvenciją $\Box(p \wedge q) \vdash \Box(p \wedge x), \Box q$.

Kaip ir kituose darbuose, kuriuose nagrinėjami sekvenciniai skaičiavimai [BL08], [HTG15], šiame darbe pateikiamos lemos, susijusios su silpninimo (*Weak*) taisyklės leistinumu. Tokios lemos dažnai pateikiamos aprašant sekvencinių skaičiavimų logines sistemas, siekiant formalizuoti struktūrinių taisyklių taikymą. Mūsų atveju šios lemos padės lengviau konstruoti tolimesnius įrodymus, kuriuose bus pasitelkiama (*Weak*) taisyklė.

5 Lema. *Taisyklė (*Weak*) yra leistina skaičiavime $LTL1$ - jeigu sekvencija S išvedama skaičiavime $LTL1$ su (*Weak*) taisyklės taikymu, tai S išvedama ir skaičiavime $LTL1$ be (*Weak*) taisyklės taikymo.*

Įrodymas.

Tarkime S yra išvedama $LTL1 + (Weak)$ ir jos išvedimo medis yra M_1 . Konstruokime naują išvedimo medį M_2 , kuriame (*Weak*) taisyklė yra, bet išvedimo medyje pakelta per vieną lygį aukšty.

Toliau nagrinėkime atvejus, kuomet medyje M_1 egzistuoja tam tikras fragmentas f_1 ir jį medyje M_2 keičiame kitu specifiniu fragmentu f_2 :

$$\text{a) } f_1 = \frac{\frac{\vdots}{(T) \frac{S_3}{S_2}}}{(Weak) \frac{S_2}{S_1}} \quad \longrightarrow \quad f_2 = \frac{(Weak) \frac{S_3}{S_1'}}{(T) \frac{S_1'}{S_1}} \quad \frac{\vdots}{S_1'}$$

Čia sekvencija S_1' gauta iš sekvencijos S_1 pritaikius (*T*) taisyklę, kur (*T*) bet kuri taisyklė išskyrus (*Weak*).

$$\text{b) } f_1 = \frac{(Weak) \frac{S_2}{S_1}}{\vdots} \quad \longrightarrow \quad f_2 = \frac{\frac{\oplus}{S_1}}{\vdots}$$

Šiuo atveju tiesiog netaikome (*Weak*) taisyklės ir S_1 taip pat yra aksioma.

$$c) \quad f_1 = \frac{(Weak) \frac{\frac{\vdots}{S_3}}{S_2}}{S_1} \longrightarrow f_2 = (Weak) \frac{\frac{\vdots}{S_3}}{S_1}$$

Vietoje dviejų iš eilės taisyklės (*Weak*) taikymų, taisyklę (*Weak*) pritaikome vieną kartą.

Gavome modifikuotą išvedimo medį, kuriame žemiausias taisyklės (*Weak*) taikymas nukeltas vienu išvedimo žingsniu aukščiau. Toliau pagal matematinę indukciją šį procesą kartojame, kol taisyklės (*Weak*) taikymas iškeliamas iki pat aksiomų. Tokiu būdu parodėme, jog sekvencija S skaičiavime $LTL1$ yra išvedama ir be taisyklės (*Weak*).

Pastaba. Atvejis a) apima ir taisyklę ($\circ$). Šis pakeitimas atrodytų taip:

$$f_1 = \frac{(Weak) \frac{(\circ) \frac{\frac{\vdots}{S_3 = \Gamma_1 \vdash \Delta_1}}{S_2 = \Pi_1, \circ\Gamma_1 \vdash \Sigma_1, \circ\Delta_1}}{S_1 = \Pi_1, \Pi_2, \circ\Gamma_1, \circ\Gamma_2 \vdash \Sigma_1, \Sigma_2, \circ\Delta_1, \circ\Delta_2}}{\vdots}$$

$$f_2 = \frac{(\circ) \frac{(Weak) \frac{\frac{\vdots}{S_3 = \Gamma_1 \vdash \Delta_1}}{S'_1 = \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}}{S_1 = \Pi_1, \Pi_2, \circ\Gamma_1, \circ\Gamma_2 \vdash \Sigma_1, \Sigma_2, \circ\Delta_1, \circ\Delta_2}}{\vdots}$$

Pirmuoju atveju (fragmente f_1) iš pradžių yra pritaikyta (*Weak*) taisyklė, o po jos ($\circ$). Antruoju atveju (fragmente f_2) šių taisyklių taikymo tvarka apsikeičia vietomis. Galutinis rezultatas (sekvencija S_3) lieka toks pats.

7.2. $LTL1$ ir klasikinio sekvencinio skaičiavimo ekvivalentumas

1 Teorema. *Sekvencija iš fragmento α_1 yra išvedama skaičiavime $LTL1$ tada ir tik tada, kai ji išvedama skaičiavime LTL .*

Pasižymėkime $\Box\Gamma = \Box G_1, \Box G_2, \dots, \Box G_k$. Π, Σ - tik atominių formulių aibės.

Irodymas.

($\Rightarrow$) Tarkime sekvencija S yra išvedama skaičiavime $LTL1$, tada parodysime, kad S išvedama ir skaičiavime LTL . Jos išvedimo medis skaičiavime $LTL1$ yra M_1 . Iš M_1 medžio konstruojame medį M_2 tokiu būdu:

1. Jeigu išvedimo medyje M_1 sutinkame taisyklę iš aibės

$$T = \{(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\circ), (\neg \vdash), (\vdash \neg), (\Box \vdash)\}$$

ši išvedimo žingsnį paliekame.

2. Taisyklės $(\vdash \Box_{L*})$ taikymo išvadą pasižymėkime $S' = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n$. Kadan-
gi sekvencija yra išvedama skaičiavime $LTL1$, turi būti išvedama viena iš OR šakų.

(a) Sakykime, kad išvedama šaka $\Box\Gamma \vdash F_i$. Jeigu išvedimo medyje sutinkame taisyklę $(\vdash \Box_{L*})$, tai šaką $B =$

$$\frac{\begin{array}{c} \vdots \\ \Box\Gamma \vdash F_1 \end{array} \parallel \dots \parallel \begin{array}{c} \vdots \\ \Box\Gamma \vdash F_i \end{array} \parallel \dots \parallel \begin{array}{c} \vdots \\ \Pi, \Box\Gamma \vdash \Sigma \end{array}}{S' = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n},$$

keičiame šaką $B' =$

$$(Weak) \frac{S_2 = \Pi, \Box\Gamma \vdash \Sigma, \Box F_i}{S' = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n}$$

Toliau taikome LTL skaičiavimo taisykles sekvencijai S_2 :

$$\begin{array}{c} \oplus - ciklinė aksioma \\ (\circ) \frac{S_2^* = \Box\Gamma \vdash \Box F_i}{\Gamma, \circ\Box\Gamma \vdash \circ\Box F_i} \\ \\ (\vdash \Box_L) \frac{\begin{array}{c} \text{(išvedama)} \\ S_1^* = \Box\Gamma \vdash F_i \end{array} \quad k * (\Box \vdash) \frac{\vdots}{\Box\Gamma \vdash \circ\Box F_i}}{(Weak) \frac{\Box\Gamma \vdash \Box F_i}{S_2 = \Pi, \Box\Gamma \vdash \Sigma, \Box F_i}} \end{array}$$

(b) Jeigu nėra tokios $\Box\Gamma \vdash F_i$ šakos, kuri būtų įrodoma, vadinasi įrodoma $\Pi, \Box\Gamma \vdash \Sigma$.

Šiuo atveju, išvedimo medžio šaką B , keičiame šaką $B'' =$

$$(Weak) \frac{\Pi, \Box\Gamma \vdash \Sigma}{S' = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n}$$

Čia $k * (T)$ reiškia, jog k kartų buvo taikyta taisyklė T . $\circ\Box\Gamma$ žymi formulių aibę, gautą k kartų pritaikius taisyklę $(\Box \vdash)$. Visuose išvedimo medžio lapuose gavome LTL skaičiavimo aksiomas arba išvedamas sekvencijas: S_1^* yra išvedama, pagal apibrėžimą, o S_2^* - ciklinė aksioma. Pakei-
tus visus taisyklės $(\vdash \Box_{L*})$ taikymus sukonstravome medį M_2 , kuriame taikomos tik taisyklės iš
skaičiavimo LTL .

($\Leftarrow$) Tarkime sekvencija S yra išvedama skaičiavime LTL , tada parodysime, kad S išvedama ir skaičiavime $LTL1$. Tegu S išvedimo medis skaičiavime LTL yra M_1 . Iš M_1 medžio konstruojame medį M_2 tokiu būdu:

1. Jeigu išvedimo medyje M_1 sutinkame taisyklę iš aibės $T = \{(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\circ), (\neg \vdash), (\vdash \neg), (\Box \vdash), (Weak)\}$ šį išvedimo žingsnį paliekame.
2. Taisyklės $(\vdash \Box_L)$ taikymo išvadą pasižymėkime $S' = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n$. Žinome, kad ši sekvencija yra išvedama skaičiavime LTL .
3. Jeigu skaičiavime LTL išvedama sekvencija S' , tai išvedama sekvencija (pagal Lemą 2):
 - a) $\Pi, \Box\Gamma \vdash \Sigma$, arba
 - b) $\Box\Gamma \vdash \Box F_1, \dots, \Box F_n$
4. Nagrinėkime atvejį 3a). Jeigu skaičiavime LTL išvedama sekvencija $\Pi, \Box\Gamma \vdash \Sigma$, tai esama išvedimo šaka keičiama į išvedimo šaką:

$$(Weak) \frac{\Pi, \Box\Gamma \vdash \Sigma}{S'}$$

Skaičiavime $LTL1$ šios sekvencijos išvedimas yra toks pats, nes sukcedente nebelieka formulų apribotų $\Box$ operatoriumi ir nereikės taikyti $(\vdash \Box_L)$ taisyklės.

5. Nagrinėkime atvejį 3b). Jeigu išvedama sekvencija $\Box\Gamma \vdash \Box F_1, \dots, \Box F_n$ taikome Lemą 4 ir gauname du atvejus:
 - a) Skaičiavime LTL išvedama $S''_1 = \Box\Gamma \vdash F_i$, arba
 - b) Skaičiavime LTL išvedama $S''_2 = \Box\Gamma \vdash$
6. Nagrinėkime atvejį 5a). Jeigu išvedama S''_1 , išvedimo medžio šaką $B =$

$$(\vdash \Box_L) \frac{\Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, F_i, \dots, \Box F_n}{S'} \frac{\Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_i, \dots, \Box F_n}{S'}$$

keičiame šaką $B' =$

$$(\vdash \Box_{L*}) \frac{\dots \parallel \frac{\text{išvedama skaičiavime } LTL1}{S''_1 = \Box\Gamma \vdash F_i} \parallel \dots}{S' = \Pi, \Box\Gamma \vdash \Sigma, \Box F_1, \dots, \Box F_n}$$

Pastaba. Kitos OR taisyklės šakos nėra reikalingos, kadangi pasirenkama išvedama sekvencija.

7. Nagrinėkime atvejį 5b). Jeigu išvedama S''_2 , išvedime nebereikalinga taikyti $(\vdash \Box)$ taisyklės ir išvedimas lieka toks pats kaip ir skaičiavime LTL .

Kadangi konstruodami medį M_2 4-ajame žingsnyje pasinaudojome taisykle (*Weak*) ir palikome visas išvedimo vietas, kuriose ši taisyklė galėjo būti taikoma pradiniam LTL išvedime, gautasis išvedimas formaliai priklauso skaičiavimui $LTL1 + (Weak)$. Vis dėlto, remiantis Lema 5, jeigu sekvencija S išvedama $LTL1 + (Weak)$, tai ji išvedama ir $LTL1$ be (*Weak*) taikymo.

7.3. $LTL1$ ir LTL skaičiavimų palyginimas

Naujame skaičiavime $LTL1$ nėra ciklinių aksiomų ir paieškos išvedimo medis supaprastėja, todėl skaičiavimas $LTL1$ yra efektyvesnis už skaičiavimą LTL fragmente α_1 . Nagrinėkime sekvencijos $S = \Box(p \wedge q) \vdash \Box(s \vee q), \Box z$ paieškos išvedimo medį abiejuose skaičiavimuose.

Pirmiausiai nagrinėkime išvedimo paieškos medį skaičiavime LTL :

$$\begin{array}{c}
\frac{S_1}{\Box(p \wedge q) \vdash s \vee q, \Box z} \quad \frac{\frac{\frac{S_3}{\Box(p \wedge q) \vdash \Box(s \vee q)}}{\circ\Box(p \wedge q), p, q \vdash \circ\Box(s \vee q), z}}{\circ\Box(p \wedge q), p \wedge q \vdash \circ\Box(s \vee q), z} \quad \frac{S_2}{\Box(p \wedge q) \vdash \circ\Box(s \vee q), \circ\Box z} \\
\hline
\Box(p \wedge q) \vdash \Box(s \vee q), \Box z
\end{array}$$

$$\begin{array}{c}
\frac{\frac{\frac{\oplus}{\circ\Box(p \wedge q), p, q \vdash s, q, z}}{\circ\Box(p \wedge q), p, q \vdash s \vee q, z}}{\circ\Box(p \wedge q), p \wedge q \vdash s \vee q, z} \quad \frac{\frac{\frac{\oplus}{\circ\Box(p \wedge q), p, q \vdash s, q, \circ\Box z}}{\circ\Box(p \wedge q), p \wedge q \vdash s, q, \circ\Box z}}{\Box(p \wedge q) \vdash s, q, \circ\Box z} \\
\hline
\Box(p \wedge q) \vdash s \vee q, \Box z
\end{array}$$

$$\frac{S_1}{\Box(p \wedge q) \vdash s \vee q, \Box z}$$

$$\frac{\frac{\frac{loop\oplus}{\Box(p \wedge q) \vdash \Box(s \vee q), \Box z}}{\circ\Box(p \wedge q), p \wedge q \vdash \circ\Box(s \vee q), \circ\Box z}}{\Box(p \wedge q) \vdash \circ\Box(s \vee q), \circ\Box z}$$

$$\frac{S_2}{\Box(p \wedge q) \vdash \circ\Box(s \vee q), \circ\Box z}$$

$$\frac{\frac{\frac{\oplus}{\circ\Box(p \wedge q), p, q \vdash s, q}}{\circ\Box(p \wedge q), p \wedge q \vdash s, q}}{\Box(p \wedge q) \vdash s, q} \quad \frac{\frac{\frac{loop\oplus}{\Box(p \wedge q) \vdash \Box(s \vee q)}}{\circ\Box(p \wedge q), p, q \vdash \circ\Box(s \vee q)}}{\circ\Box(p \wedge q), p \wedge q \vdash \circ\Box(s \vee q)} \\
\hline
\Box(p \wedge q) \vdash \Box(s \vee q)$$

$$\frac{S_3}{\Box(p \wedge q) \vdash \Box(s \vee q)}$$

Dabar nagrinėkime sekvencijos S išvedimo paieškos medį skaičiavime $LTL1$:

$$\begin{array}{c}
 \frac{\oplus}{\frac{\frac{\frac{\circ \Box(p \wedge q), p, q \vdash s, q}{\circ \Box(p \wedge q), p \wedge q \vdash s, q}}{\circ \Box(p \wedge q), p \wedge q \vdash s \vee q}}{\Box(p \wedge q) \vdash s \vee q}} \quad || \quad \frac{\frac{\frac{\frac{\text{šaka neišvedama}}{\Box(p \wedge q) \vdash}}{\vdots}}{\Box(p \wedge q) \vdash}}{\frac{\frac{\circ \Box(p \wedge q), p, q \vdash z}{\circ \Box(p \wedge q), p \wedge q \vdash z}}{\Box(p \wedge q) \vdash z}} \\
 \hline
 \Box(p \wedge q) \vdash \Box(s \vee q), \Box z
 \end{array}$$

Kadangi skaičiavime $LTL1$ nėra ciklinių aksiomų, skaičiavimas $LTL1$ fragmente α_1 tampa efektyvesnis už skaičiavimą LTL . Taip pat pastebėkime, jog šaka $\Box(p \wedge q) \vdash z$ yra neišvedama - susidaro neigiamas ciklas, tačiau skaičiavime $LTL1$ pritaikius taisyklę $(\vdash \Box_L^*)$ užtenka, jog būtų išvedama viena iš šakų.

8. Sekvenciniai skaičiavimai su apribotomis taisyklėmis

Šiame skyrelyje pristatomi du nauji sekvenciniai skaičiavimai $LTL2$ ir $LTL3$ su taisyklėmis apribotomis primarinėms sekvencijoms.

8.1. Skaičiavimas $LTL2$

Skaičiavime $LTL1$ gali atsirasti neigiami ciklai, kurių aptikimas yra būtinas gauti sekvencijos išvedimui. Tai reiškia, kad išvedimo metu privaloma taikyti ciklą aptikimą norint gauti baigtinumą, todėl toliau nagrinėsime galimybę eliminuoti ir šio tipo (neigiamus) ciklus. Šiame skyriuje pristatomas naujas sekvencinis skaičiavimas $LTL2$. Suformuojama ir įrodoma pagalbinė lema. Įrodomas ekvivalentumas tarp skaičiavimų LTL ir $LTL2$.

34 Apibrėžimas. $LTL2$ yra tiesinės laiko logikos sekvencinis skaičiavimas su taisyklėmis $(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\Box \vdash), (\vdash \Box), (\circ_p)$ ir aksioma $\Gamma, F \vdash \Delta, F$.

Skaičiavime $LTL2$, klasikinio skaičiavimo taisyklė $(\circ)$ keičiama į taisyklę $(\circ_p)$, kurią galima taikyti tik primarinei sekvencijai.

Pavyzdys. Nagrinėkime išvedimo paieškos medžio fragmentą skaičiavime $LTL2$:

$$\begin{array}{c} \vdots \\ (\circ_p) \frac{\Gamma, \circ\Box\Gamma, \Pi, \circ\Box\Pi \vdash \circ\Box\Delta}{\Gamma, \circ\Box\Gamma, \Box\Pi \vdash \circ\Box\Delta} \\ (\Box \vdash) \frac{\Gamma, \circ\Box\Gamma, \Box\Pi \vdash \circ\Box\Delta}{\Box\Gamma, \Box\Pi \vdash \circ\Box\Delta} \\ \vdots \end{array}$$

Pavyzdyje matome, jog prieš taikant taisyklę $(\circ_p)$ skaičiavimo apribojimai reikalauja prieš tai pritaikyti visas įmanomas $(\Box \vdash)$ taisykles. Be šio apribojimo, sekvencijai $\Gamma, \circ\Box\Gamma, \Box\Pi \vdash \circ\Box\Delta$ galėtų būti pritaikyta $(\circ)$ taisyklė ir gauta nauja sekvencija $\Box\Gamma \vdash \Box\Delta$.

6 Lema. Taisyklė (*Weak*) yra leistina skaičiavime $LTL2$ - jeigu sekvencija S išvedama skaičiavime $LTL2$ su (*Weak*) taisyklės taikymu, tai S išvedama ir skaičiavime $LTL2$ be (*Weak*) taisyklės taikymo.

Įrodymas. Lemos įrodymas toks pats kaip ir Lemos 5.

8.1.1. *LTL* ir *LTL2* ekvivalentumas

2 Teorema. *Sekvencija S yra išvedama skaičiavime LTL tada ir tik tada, kai ji išvedama skaičiavime $LTL2$.*

Irodymas.

Raidėmis Π, Δ žymimos baigtinės formulių aibės, kurios gali būti ir tuščios.

($\Rightarrow$) Tarkime sekvencija S yra išvedama skaičiavime LTL , tada parodysime, kad S išvedama ir skaičiavime $LTL2$. Jos išvedimo paieškos medis skaičiavime LTL yra M_1 . Iš medžio M_1 konstruojame medį M_2 tokiu būdu:

1. Jeigu medyje sutinkame taisyklę $(\circ)$, kuri buvo taikyta ne primarinei sekvencijai, tai šaką $B =$

$$(\circ) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

keičiame šaką $B' =$

$$(Weak) \frac{(\circ_p) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S'' = \circ\Delta_1 \vdash \circ\Delta_2}}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

2. Jeigu medyje sutinkame taisyklę $(\circ)$, kuri buvo taikyta primarinei sekvencijai, tai šaką $B =$

$$(\circ) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

keičiame šaką $B' =$

$$(\circ_p) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

Šiuo atveju sekvencijos šakoje liko tos pačios, tačiau pasikeitė taisyklės vardas.

3. Jeigu išvedimo medyje M_1 sutinkame taisyklę iš aibės

$$T = \{(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\vdash \Box)\}$$

šį išvedimo žingsnį paliekame.

Pakeitę visus tokius atvejus, sukonstravome medį M_2 , kuriame taisyklių taikymas atitinka leistiną taisyklių taikymą skaičiavime $LTL2 + (Weak)$, o tada pagal lemą 6 sekvencija išvedama ir skaičiavime $LTL2$.

($\Leftarrow$) Tarkime sekvencija S yra išvedama skaičiavime $LTL2$, tada parodysime, kad S išvedama ir skaičiavime LTL . Sakykime sekvencijos S išvedimo paieškos medis skaičiavime $LTL2$ yra M_1 .

Sekvencijos S išvedimo medyje M_1 visus taisyklės $(\circ_p)$ taikymus pakeičiame į taisyklės $(\circ)$ taikymus ir gauname išvedimo medį M_2 skaičiavime LTL .

8.2. Skaičiavimas $LTL3$

Šiame skyriuje pristatomas naujas sekvencinis skaičiavimas $LTL3$ fragmentui α_1 . Įrodomas ekvivalentumas tarp skaičiavimų $LTL1$ ir $LTL3$. Galiausiai pateikiamos ir įrodomos dvi pagalbinės lemos.

35 Apibrėžimas. $LTL3$ yra tiesinės laiko logikos sekvencinis skaičiavimas su taisyklėmis $(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\Box \vdash), (\vdash \Box_{L*}), (\circ_p)$ ir aksioma $\Gamma, F \vdash \Delta, F$.

Skaičiavimas $LTL3$ yra gautas iš skaičiavimo $LTL1$ tik pakeitus taisyklę $(\circ)$ į $(\circ_p)$. Taisyklę $(\circ_p)$ galima taikyti tik primarinei sekvencijai.

8.2.1. $LTL1$ ir $LTL3$ ekvivalentumas

7 Lema. Taisyklė (*Weak*) yra leistina skaičiavime $LTL3$ - jeigu sekvencija S išvedama skaičiavime $LTL3$ su (*Weak*) taisyklės taikymu, tai S išvedama ir skaičiavime $LTL3$ be (*Weak*) taisyklės taikymo.

Irodymas. Lemos įrodymas toks pats kaip ir Lemos 5.

3 Teorema. Sekvencija iš fragmento α_1 yra išvedama skaičiavime $LTL1$ tada ir tik tada, kai ji išvedama skaičiavime $LTL3$.

Irodymas.

Raidėmis Π, Δ žymimos baigtinės formulių aibės, kurios gali būti ir tuščios.

($\Rightarrow$) Tarkime sekvencija S yra išvedama skaičiavime $LTL1$, tada parodysime, kad S išvedama ir skaičiavime $LTL3$. Jos išvedimo paieškos medis skaičiavime $LTL1$ yra M_1 . Iš medžio M_1 konstruojame medį M_2 tokiu būdu:

1. Jeigu medyje sutinkame taisyklę $(\circ)$, kuri buvo taikyta ne primarinei sekvencijai, tai šaką $B =$

$$(\circ) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

keičiame šaką $B' =$

$$(\circ_p) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S'' = \circ\Delta_1 \vdash \circ\Delta_2} \\ (\text{Weak}) \frac{S'' = \circ\Delta_1 \vdash \circ\Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

2. Jeigu medyje sutinkame taisyklę $(\circ)$, kuri buvo taikyta primarinei sekvencijai, tai šaką $B =$

$$(\circ) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

keičiame šaką $B' =$

$$(\circ_p) \frac{S_1 = \Delta_1 \vdash \Delta_2}{S' = \Pi, \circ\Delta_1 \vdash \Sigma, \circ\Delta_2}$$

Šiuo atveju sekvencijos šakoje liko tos pačios, tačiau pasikeitė taisyklės vardas.

3. Jeigu išvedimo medyje M_1 sutinkame taisyklę iš aibės

$$T = \{(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\vdash \Box_{L*})\}$$

šį išvedimo žingsnį paliekame.

Pakeitę visus tokius atvejus, sukonstravome medį M_2 , kuriame taisyklių taikymas atitinka leistiną taisyklių taikymą skaičiavime $LTL3 + (Weak)$, o tada pagal lemą 7 sekvencija išvedama ir skaičiavime $LTL3$.

($\Leftarrow$) Tarkime sekvencija S yra išvedama skaičiavime $LTL3$, tada parodysime, kad S išvedama ir skaičiavime $LTL1$. Sakykime sekvencijos S išvedimo medis skaičiavime $LTL3$ yra M_1 .

Sekvencijos S išvedimo medyje M_1 visus taisyklės $(\circ_p)$ taikymus pakeičiame į taisyklės $(\circ)$ taikymus ir gauname išvedimo medį M_2 skaičiavime $LTL1$.

8.2.2. Skaičiavimo $LTL3$ savybės

Šiame skyriuje apibrėžiama ir įrodoma pagalbinė lema. Vėlesniuose skyriuose pasinaudosime šia lema siekiant įrodyti skaičiavimų ekvivalentumą.

8 Lema. *Jeigu sekvencija iš fragmento α_1 išvedama skaičiavime $LTL3$, tai egzistuoja išvedimo medis kuriame nėra $(\circ_p)$ taisyklės taikymų.*

Įrodymas. Tarkime sekvencija S išvedama skaičiavime $LTL3$. Nagrinėkime sekvencijos S išvedimo paieškos medžio M_1 fragmentą f_1 :

$$\begin{array}{c} \vdots \\ \hline S_1 = \Box A_1, \dots, \Box A_n \vdash \\ (\circ_p) \frac{S_p = \Pi', \circ \Box A_1, \dots, \circ \Box A_n \vdash \Sigma'}{S = \Pi, \Box A_1, \dots, \Box A_n \vdash \Sigma, \Box F_1, \dots, \Box F_n} \\ \vdots \\ (T) * k \frac{}{S = \Pi, \Box A_1, \dots, \Box A_n \vdash \Sigma, \Box F_1, \dots, \Box F_n} \end{array}$$

Skaičiavime $LTL3$ taisyklę $(\circ_p)$ galime taikyti tik primarinei S_p sekvencijai. Iki sekvencijos S_p buvo pritaikyta k taisyklių iš aibės $T = \{(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\vdash \Box L*), (\Box \vdash)\}$. Toliau nagrinėkime paieškos medžio fragmentą f_1 :

1. Jeigu išvedama $S_1 = \Box A_1, \dots, \Box A_n \vdash$, tai išvedama ir $S_2 = A_1, \dots, A_n \vdash$.

Taip yra todėl, kadangi sekvencijai S_1 pritaikius visas galimas modales taisykles (šiuo atveju $(\Box \vdash)$, gauname:

$$\begin{array}{c} S_3 = A_1, \dots, A_n, \circ \Box A_1, \dots, \circ \Box A_n \vdash \\ \vdots \\ (\Box \vdash) * n \frac{}{S_1 = \Box A_1, \dots, \Box A_n \vdash} \end{array}$$

Formulės $A_1, \dots, A_n$ nėra suvaržytos jokiais modalumo operatoriais (nes šios formulės yra iš fragmento α_1). Jeigu S_2 neišvedama, tai sekvencijai S_2 taikant visas logines taisykles (kurios yra apverčiamos) atsiras bent viena šaka, kuri nebus aksioma ir bus sudaryta tik iš atominių formulių. Sakykime tokia yra $S_4 = \Sigma \vdash \Pi$. Tada sekvencijai S_3 pritaikius visas logines taisykles (tas pačias ir tokia pačia tvarka kaip ir S_2) atitinkamoje medžio šakoje gausime primarinę sekvenciją $S'_4 = \Sigma, \circ \Box A_1, \dots, \circ \Box A_n \vdash \Pi$, kuri nebus aksioma ir jai bus galima taikyti tik taisyklę $(\circ_p)$. S'_4 pritaikius $(\circ_p)$ gausime sekvenciją $S''_4 = \Box A_1, \dots, \Box A_n \vdash$. Tuomet $S_1 \rightsquigarrow S''_4$ yra ciklas, bet ne ciklinė aksioma, o tai reiškia, jog sekvencijos S'_4 kaip ir S''_4 yra neišvedamos. Dėl to neišvedama ir S_3 ir S_1 (nes taikėme tik apverčiamas taisykles). Gavome prieštarą, nes turėjome sekvencijos S išvedimo medį, kuriame visos sekvencijos, taip pat ir S_1 , buvo išvedamos. Taigi, jeigu S_1 išvedama, tai išvedama ir S_2 .

2. Jeigu išvedama sekvencija $S_2 = A_1, \dots, A_n \vdash$, tai išvedama ir sekvencija S , be taisyklės $(\circ_p)$ taikymo.

Sekvencijos S išvedimo paieškos medžio M_1 fragmentą f_1 keičiame fragmentu f_2 :

$$\begin{array}{c}
\vdots \\
\hline
A_1, \dots, A_n \vdash \\
(Weak) \frac{A_1, \dots, A_n \vdash}{A_1, \dots, A_n, \circ \Box A_1, \dots, \circ \Box A_n \vdash} \\
\hline
\vdots \\
(\Box \vdash) * n \frac{\vdots}{\Box A_1, \dots, \Box A_n \vdash} \\
(Weak) \frac{(\Box \vdash) * n \frac{\vdots}{\Box A_1, \dots, \Box A_n \vdash}}{S = \Pi, \Box A_1, \dots, \Box A_n \vdash \Sigma, \Box F_1, \dots, \Box F_n}
\end{array}$$

Pakeitę visus tokius fragmentus kaip f_1 fragmentais f_2 gauname naują išvedimo medį, kuriame nebėra taisyklės $(\circ_p)$ taikymų skaičiavime $LTL3 + (Weak)$ ir pagal lemą 7 skaičiavime $LTL3$. Parodėme, skaičiavime $LTL3$ nereikalingas taisyklės $(\circ_p)$ taikymas.

9. Skaičiavimas $LTL4$

Šiame skyriuje pristatomas sekvencinis skaičiavimas tiesinėje laiko logikoje fragmentui α_1 . Įrodomas ekvivalentumas tarp skaičiavimų LTL , $LTL3$ ir $LTL4$ fragmente α_1 .

36 Apibrėžimas. $LTL4$ yra tiesinės laiko logikos sekvencinis skaičiavimas su taisyklėmis $(\vee \vdash), (\vdash \vee), (\wedge \vdash), (\vdash \wedge), (\neg \vdash), (\vdash \neg), (\Box \vdash), (\vdash \Box_{L*})$ ir aksioma $\Gamma, F \vdash \Delta, F$.

Skaičiavimas $LTL4$ yra gaunamas iš skaičiavimo $LTL3$ pašalinus taisyklę $(\circ_p)$. Taigi skaičiavime $LTL4$ visai nelieta "sekančiame žingsnyje" (*angl.* next) tipo taisyklės. Skaičiavime $LTL4$, formuojant išvedimo paieškos medį, nebereikia taikyti jokio ciklų aptikimo, kadangi išvedimo paieškos medyje netaikant $(\circ)$ ar $(\circ_p)$ taisyklių nebeįmanoma gauti tokios pačios sekvencijos, todėl ir ciklo.

9.1. $LTL3$ ir $LTL4$ ekvivalentumas

4 Teorema. Sekvencija iš fragmento α_1 yra išvedama skaičiavime $LTL3$ tada ir tik tada, kai ji išvedama skaičiavime $LTL4$.

Įrodymas.

($\Rightarrow$) Tarkime sekvencija S yra išvedama skaičiavime $LTL3$, tada ji išvedama šiame skaičiavime ir be taisyklės $(\circ_p)$ taikymo (pagal Lemą 8). Išvedimo paieškos medis skaičiavime $LTL3$ be taisyklės $(\circ_p)$ taikymo ir yra išvedimo paieškos medis skaičiavime $LTL4$.

($\Leftarrow$) Tarkime sekvencija S yra išvedama skaičiavime $LTL4$ ir jos išvedimo medis yra M_1 , tada sekvencijos S išvedimo medis skaičiavime $LTL3$ ir bus M_1 .

9.2. LTL ir $LTL4$ ekvivalentumas

5 Teorema. Skaičiavimai LTL ir $LTL4$ yra ekvivalentūs fragmente α_1 .

Įrodymas. Kadangi žinome, jog $LTL4 \iff LTL3$, $LTL3 \iff LTL1$ ir $LTL1 \iff LTL$ fragmente α_1 , taigi $LTL \iff LTL4$ fragmente α_1 .

Šiame sekvenciniame skaičiavime visiškai eliminuojamas ciklų atsiradimas (tiek ciklinių aksiomų, tiek neigiamų ciklų), t.y. skaičiavimas $LTL4$ yra beciklis fragmente α_1 , todėl šis skaičiavimas yra efektyvesnis tiek už LTL , tiek už $LTL1$ fragmente α_1 .

Nagrinėkime pavyzdį sekvencijos $\Box(p \wedge q) \vdash \Box(s \vee q), \Box z$ išvedimo pavyzdį skaičiavime $LTL4$:

$$\begin{array}{c}
 \oplus \\
 \hline
 \frac{\circ\Box(p \wedge q), p, q \vdash s, q}{\circ\Box(p \wedge q), p \wedge q \vdash s, q} \\
 \hline
 \frac{\circ\Box(p \wedge q), p \wedge q \vdash s \vee q}{\Box(p \wedge q) \vdash s \vee q}
 \end{array}
 \quad
 \begin{array}{c}
 \text{šaka neišvedama} \\
 \hline
 \frac{\circ\Box(p \wedge q), p, q \vdash z}{\circ\Box(p \wedge q), p \wedge q \vdash z} \\
 \hline
 \Box(p \wedge q) \vdash z
 \end{array}
 \quad
 \parallel
 \quad
 \frac{\Box(p \wedge q) \vdash s \vee q \quad \Box(p \wedge q) \vdash z}{\Box(p \wedge q) \vdash \Box(s \vee q), \Box z}$$

Pastebėjime, jog pirma išvedimo šaka, kaip ir skaičiavime $LTL1$, yra išvedama. Tačiau, skirtingai nei skaičiavime $LTL1$, išvedinėjant nebereikia taikyti ciklų aptikimo. Dešinėje šakoje gavome sekvenciją, kuri nėra aksioma ir šiai sekvencijai negalima pritaikyti jokios taisyklės. Pavyzdyje kairėje šakoje ciklas nesusidaro, nei $LTL1$, nei $LTL4$ skaičiavimuose, tačiau dešinėje šakoje $LTL1$ skaičiavime susidaro ciklas:

$$\begin{array}{c}
 \oplus \\
 \hline
 \frac{\circ \Box(p \wedge q), p, q \vdash s, q}{\circ \Box(p \wedge q), p \wedge q \vdash s, q} \\
 \hline
 \frac{\circ \Box(p \wedge q), p \wedge q \vdash s \vee q}{\Box(p \wedge q) \vdash s \vee q} \quad || \quad \frac{\text{šaka neišvedama}}{\Box(p \wedge q) \vdash} \\
 \hline
 \frac{\Box(p \wedge q) \vdash s \vee q \quad \Box(p \wedge q) \vdash \Box(s \vee q), \Box z}{\Box(p \wedge q) \vdash \Box(s \vee q), \Box z}
 \end{array}$$

Nebandant aptikti ciklų, šios šakos išvedimas būtų begalinis. Skaičiavime $LTL4$ ciklas susidaryti negali, nes sekvencijai $\Box(p \wedge q) \vdash \Box(s \vee q), \Box z$ nebegalima taikyti jokios $LTL4$ skaičiavimo taisyklės.

9.3. $LTL4$ ir LTL skaičiavimų palyginimas

Skaičiavime $LTL4$ visiškai eliminuotas „*next*“–tipo ($\circ$) operatoriaus taisyklių taikymas, todėl išvedimo paieškos medyje nelieta jokių ciklų – nei „gerųjų“ (ciklinių aksiomų), nei „blogųjų“ (neigiamų ciklų). Ciklų aptikimas LTL skaičiavime sudaro didžiausią automatinės įrodymų paieškos kaštų dalį, todėl jų pašalinimas įgalina mažesnę išvedimo paieškos medį ir paprastesnę automatizaciją.

$LTL4$ išlaiko pilnumą ir korektiškumą α_1 fragmente ir tampa visiškai *becikliu* skaičiavimu, todėl yra efektyvesnis (lyginant su skaičiavimu LTL) fragmente α_1 tiek automatizuotam, tiek rankiniam išvedimui.

10. Rezultatai ir išvados

Šiame darbe buvo nagrinėjama tiesinė laiko logika, įrodytos tam tikros savybės, kurios leido apibrėžti logikos fragmentą α_1 bei tam fragmentui sukonstruoti efektyvesnius skaičiavimus, iš kurių skaičiavimas $LTL4$ visai nereikalauja ciklų aptikimo. Skaičiavimas tampa efektyvesnis už klasikinį tiesinės laiko logikos sekvencinį skaičiavimą LTL minėtame fragmente. Pagrindiniai šio darbo **rezultatai** yra:

1. Apibrėžtas unarinis tiesinės laiko logikos fragmentas α_1 , kuriame limituojamas formulių suvaržymas būtinumo operatoriais.
2. Apibrėžtos ir įrodytos pagalbinės lemos skaičiavimui LTL , kurios leidžia pasinaudoti tam tikromis LTL skaičiavimo savybėmis. Lemomis įrodoma, jog jeigu skaičiavime LTL išvedamos tam tikro tipo sekvencijos, vadinasi turi būti išvedamos ir kitos specifinės sekvencijos. Įrodytos lemos panaudotos ekvivalentumo tarp skaičiavimų LTL ir $LTL1$ įrodyme.
3. Sukurtas naujas tiesinės laiko logikos sekvencinis skaičiavimas $LTL1$, kuriame nėra ciklinių aksiomų. Sugalyvota nauja taisyklė (OR tipo), leidžianti nagrinėti tik vieną iš taisyklės prielaidų - likusios išvedimo paieškos medžio sekvencijos gali būti toliau nebeišvedinėjamos. Naujos ($\vdash \Box_{L*}$) taisyklės pagalba eliminuojamos ciklinės aksiomos ir išvedimo paieškos medis tampa mažesnis.
4. Įrodytas ekvivalentumas tarp skaičiavimų LTL ir $LTL1$ fragmente α_1 . Įrodymas buvo grįstas išvedimo paieškos medžio viename skaičiavime transformavimu į išvedimo paieškos medį kitame skaičiavime.
5. Sukurtas naujas sekvencinis skaičiavimas $LTL2$ su apribota taisykle ($\circ_p$). Įrodytas ekvivalentumas tarp skaičiavimų LTL ir $LTL2$.
6. Sukurtas naujas sekvencinis skaičiavimas $LTL3$ su ($\circ_p$) ir ($\vdash \Box_{L*}$) taisyklėmis. Įrodytas ekvivalentumas tarp skaičiavimų $LTL1$ ir $LTL3$ fragmente α_1 .
7. Apibrėžtos ir įrodytos pagalbinės lemos skaičiavimams $LTL1$, $LTL2$ ir $LTL3$.
8. Sukurtas naujas sekvencinis skaičiavimas $LTL4$ be ($\circ$) ir ($\circ_p$) taisyklių, kuriame nėra ciklinių aksiomų ir neigiamų ciklų. Įrodytas ekvivalentumas tarp skaičiavimų LTL ir $LTL4$ fragmente α_1 .

Pagrindinės šio darbo **išvados**:

1. Fragmentų nagrinėjimas tiesinei laiko logikai pasiteisino - pavyko sukonstruoti efektyvesnį ir labiau automatizavimui tinkamą skaičiavimą $LTL4$ fragmente α_1 .
2. $LTL1$ skaičiavime sekvencijų išvedimas fragmente α_1 yra efektyvesnis lyginant su skaičiavimu LTL , nes konstruojamas mažesnis išvedimo paieškos medis ir tuo pačiu nebereikia tikrinti ciklinių aksiomų. Skaičiavime $LTL1$ gali atsirasti ciklai, kurie nėra ciklinės aksiomos, todėl ciklų tikrinimo eliminavimas išvedimo medžio konstravimo procese nėra galimas.

3. $LTL4$ skaičiavime sekvencijų išvedimas fragmente α_1 yra efektyvesnis nei $LTL1$, kadangi skaičiavimo $LTL4$ išvedime negali susidaryti jokie ciklai – nei ciklinės aksiomos, nei neigiami ciklai – todėl ciklų aptikimo visai nereikia taikyti.

Efektyvesnis skaičiavimas leidžia potencialiai greičiau spręsti problemas, kurios yra susijusios su sritimis, kuriose pritaikoma tiesinė laiko logika. Pavyzdžiui, programų specifikavimas ir verifikavimas, laikinosios duomenų bazės, paskirstytos ir multi-agentų sistemos, bei žinių reprezentacija ir samprotavimas. Skaičiavime $LTL1$ vis dar gali atsirasti neigiami ciklai, lemiantys šakos neišvedamumą. Skaičiavimas $LTL1$ yra efektyvesnis už skaičiavimą LTL , nes jame nėra ciklinių aksiomų, tačiau vis dar gali atsirasti neigiami ciklai. Šią skaičiavimo $LTL1$ problemą pavyko visiškai išspręsti sukonstravus skaičiavimą $LTL4$. Skaičiavime $LTL4$ negali susidaryti jokie ciklai (nei teigiami, nei neigiami), todėl išvedimo paieškos algoritmams paremtiems skaičiavimu $LTL4$ visai nereikia taikyti ciklų aptikimo, o tai daro išvedimą ne tik efektyvesnį, bet ir paprastesnį.

Literatūra

- [AF94] A. Artale ir E. Franconi. A Computational Account for a Description Logic of Time and Action. *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, p.p. 3–14, San Francisco, USA. Morgan Kaufmann Publishers Inc., 1994.
- [AP13] R. Alonderis ir R. Pliuškevičius. Sequent Systems for PLTL. *Lietuvos matematikos rinkinys*, p.p. 1–5, Vilnius, Lithuania. Vilniaus universiteto leidykla, 2013.
- [APP⁺20] R. Alonderis, R. Pliuškevičius, A. Pliuškevičienė ir H. Giedra. Loop-Type Sequent Calculi for Temporal Logic. *Journal of Automated Reasoning*, 64, 2020-12. DOI: 10.1007/s10817-020-09544-1.
- [Arz21] S. Arzaghi. An Introduction To Linear Temporal Logic (LTL). https://www.researchgate.net/publication/349145426_An_Introduction_To_Linear_Temporal_Logic_LTL, 2021. 374 KB.
- [AZ07] B. Aaron ir M. Zohar. *The Calculus of Computation: Decision Procedures with Applications to Verification*. Springer, Germany, Leipzig, 2007. 70 psl.
- [Bir07] Adomas Birštunas. Efficient loop-check for multimodal KD45n logic. *Lietuvos matematikos rinkinys*, tom. 47 numeris special issue, p.p. 351–355, Vilnius, Lithuania. Vilniaus universiteto leidykla, 2007.
- [Bir10] A. Birštunas. *Sequent calculi with an efficient loop-check for BDI logics*. Disertacija, Vilniaus universitetas, 2010.
- [BL08] Kai Brännler ir Martin Lange. Cut-free sequent systems for temporal logic. *Journal of Logic and Algebraic Programming*, 76:216–225, 2008. DOI: 10.1016/j.jlap.2008.02.004. URL: <https://doi.org/10.1016/j.jlap.2008.02.004>.
- [Cho94] J. Chomicki. Temporal query languages: A survey. Dov M. Gabbay ir Hans Jürgen Ohlbach, redaktoriai, *Temporal Logic*, p.p. 506–534, Berlin, Germany. Springer: Berlin Heidelberg, 1994.
- [FHM⁺95] R. Fagin, J. Halpern, Y. Moses ir M. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
- [HTG15] Zhé Hóu, Alwen Tiu ir Rajeev Goré. A Labelled Sequent Calculus for BBI: Proof of Theory and Proof Search. <https://arxiv.org/abs/1302.4783>, 2015. arXiv:1302.4783 [cs.LO], version 5.
- [Lam93] L. Lamport. The Temporal Logic of Actions. *ACM Transactions on Programming Languages and Systems*, 16(3):119–136, 1993.
- [MP91] Z. Manna ir A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer: New York, 1991.
- [Nor04] S. Norgėla. *Matematinė logika*. TEV, Vilnius, 2004. 191 pages.

- [Pli08] R. Pliuškevičius. Loop-free verification of termination of derivation for a fragment of dynamic logic. *Lietuvos matematikos rinkinys*, p.p. 283–287, Vilnius, Lithuania. Vilniaus universiteto leidykla, 2008.
- [Pnu06] A. Pnueli. Applications of temporal logic to the specification and verification of reactive systems: A survey of current trends. *Lecture Notes in Computer Science - LNCS*, p.p. 510–584, 2006-04. DOI: 10.1007/BFb0027047.
- [Sch93] K. Schild. Combining Terminological Logics with Tense Logic. *Portuguese Conference on Artificial Intelligence*, 1993.
- [TD16] J. Tumova ir D. Dimarogonas. Multi-agent planning under local LTL specifications and event-based synchronization. *Automatica*, 70:239–248, 2016.
- [Val80] M.K. Valiev. Decision complexity of variants of propositional dynamic logic. *Automatica*, 88:656–664, 1980.
- [Wol85] P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, 28:119–136, 1985.