



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

Sekvencinių skaičiavimų BDI logikai efektyvumo tyrimas
Research on effectiveness of sequent calculi for BDI logic

Baigiamasis magistro darbas

Atliko: Bernardas Raginis

VU e-paštas: bernardas.raginis@mif.stud.vu.lt

Vadovas: asist. dr. Adomas Birštunas

Recenzentas: asist. dr. Julius Andrikonis

Vilnius – 2025

Santrauka

Šio magistro baigiamojo darbo sritis - sekvenciniai skaičiavimai BDI logikai. Darbo tema yra "Sekvencinių skaičiavimų BDI logikai efektyvumo tyrimas". Šio darbo tikslas – apibrėžti apribojimus ciklą susidarymui, remiantis taisyklių skirtingiems modaliniams operatoriams sąveika, ir šitaip padidinti ciklą aptikimo efektyvumą bei, remiantis apibrėžtais apribojimais, sukurti sekvencijos išvedamumo patikrinimo procedūrą. Šiame darbe nagrinėjama modalumo operatorių (taisyklių šiems operatoriams taikymų) tarpusavio sąveika, suformuluojami ir įrodomi apribojimai ciklą susidarymui nagrinėjamame skaičiavime. Remiantis apribojimais apibrėžiamas efektyvesnis sekvencinis skaičiavimas su procedūra, leidžiantis patikrinti ar sekvencija yra išvedama.

Raktiniai žodžiai: sekvencija, sekvenciniai skaičiavimai, BDI logika, modalinė logika, modalumo operatorius, ciklas.

Summary

The area of this master's paper is sequent calculi for BDI logic. The more specific topic is "Research on effectiveness of sequent calculi for BDI logic". The objective of this work is to define loop formation constraints based on the interaction between rules for different modal operators and improve the efficiency of loop-check. The defined constraints should be used to create a procedure for sequent derivability check. In this work, interactions between modal operators (modal operator rule applications) are analyzed. Loop formation constraints for sequent calculus under consideration are formulated and proven. Based on the identified constraints, a more effective sequent calculus with a procedure for determining sequent derivability is defined.

Keywords: sequent, sequent calculi, BDI logic, modal logic, modal operator, loop.

Turinys

IVADAS.....	6
1. Loginiai skaičiavimai.....	7
1.1. Hilberto tipo teiginių skaičiavimas	7
1.2. Sekvencinis skaičiavimas.....	8
2. Modalinės logikos.....	10
2.1. Modalinių logikų klasifikacija	10
2.2. Laiko logika	11
2.3. Įsitikinimų logika	12
2.4. Troškimų ir ketinimų logikos.....	13
3. BDI logika.....	14
3.1. Agentai	14
3.2. „Belief–Desire–Intention“ modelis.....	14
3.3. Agentų veikimo principai BDI logikoje	15
4. Sekvenciniai skaičiavimai modalinėms logikoms	19
4.1. Ciklai sekvenciniuose skaičiavimuose modalinėms logikoms	19
4.2. Sekvencinis skaičiavimas BDI logikai SBDI*	21
4.3. Sekvencinis skaičiavimas BDI logikai SBDI su procedūra	22
4.3.1. Procedūra skaičiavimui SBDI.....	24
5. Skaičiavimo SBDI savybės.....	27
5.1. Skaičiavimų SBDI* ir SBDI ekvivalentumas.....	27
5.2. Skaičiavimo SBDI baigtinumas.....	31
6. Apribojimai ciklams skaičiavime SBDI	33
6.1. Apribojimai troškimų ir ketinimų ciklams.....	33

6.2.	Laiko ir įsitikinimų ciklai.....	37
6.3.	Apribojimai įsitikinimų ir laiko ciklams.....	39
6.4.	Sekvencinis skaičiavimas SBDI1 su procedūra.....	44
6.5.	Sekvencijų išvedimo ir ciklų paieškos pavyzdžiai skaičiavime SBDI1	47
7.	Rezultatai ir išvados.....	51
	Literatūra	53

IVADAS

Vienas iš modelių, padedančių kurti sudėtingas autonomiškas programas, yra agentai. Tai esybės, pasižyminčios tam tikromis savybėmis, iš kurių svarbiausios yra nepriklausomas veikimas tam tikroje aplinkoje ir tikslų siekimas. Protingų agentų (*angl., intelligent agent*) kūrimas yra svarbus dirbtinio intelekto srityje.

Agentų autonomiškumas leidžia jiems sėkmingai veikti besikeičiančioje aplinkoje, tačiau esminė savybė, leidžianti juos laikyti protingais, yra gebėjimas siekti tikslų [Woo00]. Vienas iš žinomų protingų agentų modelių yra BDI (Belief-Desire-Intention) modelis. Šiame modelyje agento veikimą lemia įsitikinimai, troškimai ir ketinimai. Šios charakteristikos keičiasi einant laikui ir agentui gaunant naują informaciją iš aplinkos bei vykdant neprižiūrimą mąstymo procesą.

BDI agentų atveju mąstymo procesas dažnai įgyvendinamas vykdant sekvencinius skaičiavimus. Agentas iš turimos žinių bazės periodiškai bando išvesti formules, o šių formulių išvedamumas ar neišvedamumas lemia tolimesnius agento veiksmus.

Dėl BDI logikoje naudojamų skirtingų modalumų agento vykdomi sekvenciniai skaičiavimai nebūtinai pasižymi baigtinumu. Sąveikaujant skirtingiems modalumo operatoriams gali susidaryti ciklai, dėl kurių skaičiavimą įgyvendinanti programa gali nebaigti darbo. Taigi labai svarbu yra gebėti išvengti ciklų susidarymo, o ciklams susidarius gebėti juos efektyviai identifikuoti.

Magistro baigiamojo darbo tikslas: sukurti sekvencijos išvedamumo patikrinimo procedūrą su efektyviu ciklų aptikimu BDI logikai. Šiame darbe gilinamasi į taisyklių modaliniams operatoriams taikymus ir ypač į jų tarpusavio sąveiką bei įtaką ciklų susidarymo galimumui. Bandoma atrasti sąlygas, kurios lemia ciklų susidarymo apribojimus. Siekiama, kad automatinio išvedimo sudėtingumas, taigi, ir skaičiavimą vykdančios programos vykdymo laikas būtų sumažinti.

Šiame darbe pateikiama aktuali teorinė medžiaga, susijusi su sekvenciniais skaičiavimais BDI logikai, įvairių ciklų skaičiavime BDI logikai pavyzdžiai, iliustruojantys problematiką ir darbe pasiektus rezultatus. Darbe suformuluojami ir įrodomi apribojimai ciklų susidarymui skaičiavime BDI logikai. Apibrėžiamas naujas skaičiavimas su procedūra, besiremiantis suformuluotais apribojimais. Apribojimai ir jų įtaką taip pat iliustruojami konkrečiais pavyzdžiais.

1. Loginiai skaičiavimai

1.1. Hilberto tipo teiginių skaičiavimas

Šiame poskyryje apibrėžiamas Hilberto tipo teiginių skaičiavimas. Kaip nurodoma [Nor04], tai pataisytas ir papildytas konjunkcijos ir disjunkcijos aksiomomis G. Frege 1879 m. aprašytas skaičiavimas. Vokiečių matematikas D. Hilbertas taip pat nagrinėjo tokius skaičiavimus ir pasiekė reikšmingų rezultatų, todėl įprastai tokie skaičiavimai vadinami Hilberto tipo skaičiavimais.

Hilberto tipo teiginių skaičiavimą nusako aksiomos ir išvedimo taisyklė. Aksiomų (schemų) sąrašas pateikiamas žemiau:

$$1.1 \ F \rightarrow (G \rightarrow F)$$

$$1.2 \ (F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$$

$$2.1 \ (F \& G) \rightarrow F$$

$$2.2 \ (F \& G) \rightarrow G$$

$$2.3 \ (F \rightarrow G) \rightarrow ((F \rightarrow H) \rightarrow (F \rightarrow (G \& H)))$$

$$3.1 \ F \rightarrow (F \vee G)$$

$$3.2 \ F \rightarrow (F \vee G)$$

$$3.3 \ (F \rightarrow H) \rightarrow ((G \rightarrow H) \rightarrow ((F \vee G) \rightarrow H))$$

$$4.1 \ (F \rightarrow G) \rightarrow (\neg G \rightarrow \neg F)$$

$$4.2 \ F \rightarrow \neg \neg F$$

$$4.3 \ \neg \neg F \rightarrow F$$

Hilberto tipo teiginių skaičiavime yra tik viena išvedimo taisyklė (Modus Ponens):

$$\bullet \text{ (MP)} \quad \frac{F \quad F \rightarrow G}{G}$$

Išvedimo taisyklės viršuje esančios formulės yra prielaidos (gali būti viena arba dvi), o apačioje – išvada (visada viena).

Apibrėžimas 1. Formulės **F** įrodymu Hilberto tipo teiginių skaičiavime vadiname baigtinę formulių seką, kuri baigiasi formule **F**, ir kurioje kiekviena formulė yra arba aksioma, arba gauta iš prieš ją esančių formulių pagal skaičiavimo taisyklę (Modus Ponens taisyklę teiginių logikos atveju).

1.2. Sekvencinis skaičiavimas

Šiame poskyryje apibrėžiamas sekvencinis skaičiavimas teiginių logikai. Kaip nurodoma [Nor04], tai G. Gentzeno 1930 m. aprašytas skaičiavimas. Šis skaičiavimas pranašesnis už Hilberto tipo skaičiavimą, nes leidžia įrodyti formulės neįrodomumą. Taip pat šiame skaičiavime išvedimo paieška dažnai paprastesnė (trumpesnė).

Apibrėžimas 2. Sekvencija vadiname reiškinių $F_1, \dots, F_n \vdash G_1, \dots, G_m$; čia F_i ($i = 1, \dots, n$) bei G_i ($i = 1, \dots, m$) yra formulės ir $n + m > 0$.

Raidėmis $\Gamma, \Gamma', \Gamma'', \Delta, \Delta', \Delta''$ žymimos baigtinės formulių aibės. Jos gali būti ir tuščios. Sekvencijoje $\Gamma \vdash \Delta$ aibė Γ vadinama antecedentu, o Δ — sukcedentu.

Sekvenciniame skaičiavime teiginių logikai egzistuoja vienintelė **aksioma**: $\Gamma, F \vdash \Delta, F$ ir taisyklės:

$(\rightarrow \vdash)$	$\frac{\Gamma \vdash \Delta, F \quad \Gamma, G \vdash \Delta}{\Gamma, F \rightarrow G \vdash \Delta}$	$(\vdash \rightarrow)$	$\frac{\Gamma, F \vdash \Delta, G}{\Gamma \vdash \Delta, F \rightarrow G}$
$(\& \vdash)$	$\frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \& G \vdash \Delta}$	$(\vdash \&)$	$\frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \& G}$
$(\vee \vdash)$	$\frac{\Gamma, F \vdash \Delta \quad \Gamma, G \vdash \Delta}{\Gamma, F \vee G \vdash \Delta}$	$(\vdash \vee)$	$\frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}$
$(\neg \vdash)$	$\frac{\Gamma \vdash \Delta, F}{\Gamma, \neg F \vdash \Delta}$	$(\vdash \neg)$	$\frac{\Gamma, F \vdash \Delta}{\Gamma \vdash \Delta, \neg F}$
$(W \vdash)$	$\frac{\Gamma \vdash \Delta}{F, \Gamma \vdash \Delta}$	$(\vdash W)$	$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, F}$
$(C \vdash)$	$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}$	$(\vdash C)$	$\frac{\Gamma \vdash F, F, \Delta}{\Gamma \vdash F, \Delta}$
$(P \vdash)$	$\frac{\Gamma', F, G, \Gamma'' \vdash \Delta}{\Gamma, G, F, \Gamma'' \vdash \Delta}$	$(\vdash P)$	$\frac{\Gamma \vdash \Delta', F, G, \Delta''}{\Gamma \vdash \Delta', G, F, \Delta''}$

Šios taisyklės gali būti skirstomos į logines ir struktūrines. Loginės taisyklės papildo taisyklės išvados sekvenciją loginiu operatoriumi (pvz. $(\rightarrow \vdash)$ papildo sekvenciją implikacijos operatoriumi), o struktūrinės taisyklės keičia sekvencijų struktūrą (formulių sekas) ir ignoruoja sekvencijų formulių vidinę struktūrą (pvz. $(P \vdash)$ sukeičia dvi formules vietomis nepriklausomai nuo tų formulių struktūrų).

Apibrėžimas 3. Sekvencijos S išvedimu (išvedimo medžiu) sekvenciniame skaičiavime vadinamas medis, kurio visose galinėse viršūnėse (lapuose) yra sekvencijos-aksiomos, likusiose viršūnėse — sekvencijos, gautos pagal kurią nors sekvencinio skaičiavimo taisyklę iš tiesiogiai virš jų medyje esančių formulių, ir šaknyje esanti sekvencija yra S .

2. Modalinės logikos

Modalinė logika – tai logika, praplėsta modalumo operatoriais ir atitinkamomis aksiomomis.

Modalumo operatoriai techniškai nesiskiria nuo įprastų loginių operatorių. Jie taip pat priima loginius kintamuosius ir priklausomai nuo jų reikšmių grąžina vieną iš dviejų reikšmių: *tiesa* arba *netiesa*. Šie operatoriai išplečia įprastinę logiką ir leidžia nagrinėti tikėtinumą, pastovumą ir kitus modalumus, kurių nėra galimybės išreikšti be šių operatorių.

Apibrėžimas 4. Monomodalinė logika – tai modalinė logika, kurioje naudojamas tik vienas modalumas (modalinis operatorius).

Apibrėžimas 5. Multimodalinė logika – tai modalinė logika, kurioje naudojamas daugiau nei vienas modalumas (taigi, ir daugiau nei vienas modalinis operatorius).

2.1. Modalinių logikų klasifikacija

Modalinės logikos klasifikuojamos pagal jose naudojamus modalumo operatorius (modalumus). Operatorių apibrėžimams naudojamos Hilberto tipo skaičiavimo aksiomos ir taisyklės. Šio poskyrio vėlesniuose skirsniuose pateikiami modalumo logikų, aktualių šiame darbe, pavyzdžiai bei šias logikas apibrėžiančios Hilberto tipo skaičiavimo aksiomos ir taisyklės.

Svarbu paminėti, kad skirtingose srityse taikomos skirtingos modalinės (monomodalinės ar multimodalinės) logikos. Taip pat galima situacija, kai naudojami keli tokie patys modalumai. Pavyzdžiui, kai logikoje nagrinėjami daugelio agentų įsitikinimai, įsitikinimų modaliniai operatoriai žymimi indeksais: $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \dots, \mathbf{B}_n$. Čia $\mathbf{B}_i$ - i-ojo agento įsitikinimus žymintis modalinis operatorius.

Žemiau pateikiamos Hilberto tipo skaičiavimų modalinės aksiomos, naudojamos apibrėžiant įvairias modalines logikas:

$$\text{K. } \Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$$

$$\text{D. } \Box F \rightarrow \Diamond F$$

$$4. \Box F \rightarrow \Box \Box F$$

$$5. \Diamond F \rightarrow \Box \Diamond F$$

$$\text{T. } \Box F \rightarrow F$$

$$\text{B. } F \rightarrow \Box \Diamond F$$



Pav. 1. Modalinių logikų kubas. [NZ00]

Aukščiau (Pav. 1) pateiktas modalinių logikų sąryšius vaizduojantis kubas. Šis brėžinys interpretuojamas taip:

- Ašys T, 4, 5 ir B reiškia atitinkamų aksiomų pridėjimą. Pavyzdžiui, logika K5 yra gaunama prie logikos K pridedant aksiomą 5.
- Einant į dešinę ir aukštin pasiekiamas logikos praplėtimas. Taigi, modalinė logika S4 yra praplėsta modalinė logika D4, kitaip sakant, D4 logikoje įrodomų formulių aibė yra S4 logikoje įrodomų formulių aibės griežtas poaibis.

Vėlesniuose poskyriuose glaustai apibrėžiamos šiam darbui aktualios modalinės logikos.

2.2. Laiko logika

Laiko logika yra modalinė logika, kurią naudojant samprotaujama apie laiką. Laiko logika gali būti skaidi (skaidaus laiko) arba tiesinė. Skaidžios laiko logikos atveju laikas išsišakoja, yra daug laiko linijų. Šiame darbe nagrinėjama BDI logika su tiesine laiko logika, todėl ji detaliau aprašoma žemiau.

Tiesinė laiko logika, kaip nurodoma [EH86], yra modalumo logika, nusakanti neišsišakojantį laiką. Tiesinėse laiko logikose egzistuoja vienintelė ateitis, taigi laikas yra tiesinis. Šiose modalumo logikose naudojami 4 modalumo operatoriai: $\circ$, $\square$, $\diamond$, U . Jie apibrėžiami taip:

- Formulė $\circ F$ teisinga, jei F teisinga kitu laiko momentu.
- Formulė $\Box F$ teisinga, jei F teisinga visais tolimesniais laiko momentais.
- Formulė $\Diamond F$ teisinga, jei F teisinga kuriuo nors tolimesniu laiko momentu.
- Formulė FUG teisinga, jei F teisinga bent iki momento, kai teisinga G .

Dažniausiai sutinkamose tiesinėse laiko logikose naudojami operatoriai $\Box$ ir $\circ$. $\Diamond$ operatorius dažnai nenaudojamas, nes jį galima išreikšti per $\Box$ ir $\neg$: $\Diamond F \sim \neg \Box \neg F$. Žodžiais šių operatorių sąryšį galima nusakyti šitaip: F teisinga kuriuo nors tolimesniu laiko momentu, jei nėra taip, kad visais laiko momentais F nėra teisinga.

Taip pat yra laiko logikų, kuriuose naudojamas ‘until’ operatorius U vietoj $\Box$ ir $\circ$. Nėra žinomų plačiai naudojamų laiko logikų, kuriose būtų naudojami visi keturi operatoriai kartu.

Apibrėžimas 6. Kaip nurodoma [Alo12], **Hilberto tipo skaičiavimas tiesinei laiko logikai** (naudojančiai $\Box$ ir $\circ$ operatorius) - tai skaičiavimas, kuriame yra Hilberto tipo teiginių skaičiavimo aksiomos 1.1 – 4.3, taip pat aksiomos:

- $\circ \neg F \rightarrow \neg \circ F$,
- $\circ(F \rightarrow G) \rightarrow (\circ F \rightarrow \circ G)$,
- $\Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$,
- $\Box F \rightarrow F$,
- $\Box F \rightarrow \circ \Box F$
- $(F \& \Box(F \rightarrow \circ F)) \rightarrow \Box F$,

Ir taisyklės:

$$(\circ) \quad \frac{F}{\circ F}$$

$$(\Box) \quad \frac{F}{\Box F}$$

$$(MP) \quad \frac{F \quad F \rightarrow G}{G}$$

2.3. Įsitikinimų logika

Įsitikinimų (tikėjimo) logika yra modalinė logika, kurioje samprotaujama apie tikėjimą (įsitikinimus). Kaip nurodoma [DHH15], tikėjimo logikos dar vadinamos doksastinėmis (nuo graikiško žodžio *doxa*). Tikėjimo modalumo logikoje yra vienas modalumo operatorius dažniausiai žymimas **B** (belief). Šis operatorius apibrėžiamas taip:

- **BF** reiškia, kad yra *tikima*, kad F yra teisinga.

Apie tikėjimą samprotaujama naudojant modalinę logiką KD45. Žemiau pateikiamas Hilberto tipo skaičiavimo šiai modalinei logikai apibrėžimas.

Apibrėžimas 7. Hilberto tipo skaičiavimas modalinei logikai KD45 yra skaičiavimas, kuriame yra aksiomos 1.1 - 4.3 ir K, D, 4, 5. Taip pat taisyklės:

$$(\Box) \frac{F}{\Box F}$$

$$(MP) \frac{F \quad F \rightarrow G}{G}$$

Samprotaujant apie įsitikinimus vietoje apibrėžime pateikiamo modalinio operatoriaus $\Box$ naudojamas aukščiau minimas modalinis operatorius **B**.

2.4. Troškimų ir ketinimų logikos

Troškimų ir ketinimų modalinės logikos yra logikos, kuriose samprotaujama apie troškimus ir ketinimus. Šiose modalumo logikose yra po vieną modalumo operatorių - **D** (desire) troškimų modalinėje logikoje ir **I** (intention) ketinimų modalinėje logikoje. Šie operatoriai apibrėžiami taip:

- **DF** reiškia, kad yra norima/trokštama, kad F būtų teisinga.
- **IF** reiškia, kad yra ketinama padaryti, kad F būtų teisinga.

Tiek apie troškimus, tiek apie ketinimus samprotaujama naudojant modalinę logiką KD. Taigi šie modalumai yra ekvivalentūs (juos nusako tos pačios loginės taisyklės), todėl jie pateikiami viename poskyryje. Nepaisant to, kad šie modalumai yra ekvivalentūs, šio darbo (ir BDI logikos) kontekste jie išskiriami ir žymimi skirtingais operatoriais, kad būtų galima atskirti agento troškimus nuo ketinimų.

Žemiau pateikiamas Hilberto tipo skaičiavimo modalinei logikai KD apibrėžimas.

Apibrėžimas 8. Hilberto tipo skaičiavimas modalinei logikai KD yra skaičiavimas, kuriame yra aksiomos 1.1 - 4.3 ir K, D. Taip pat taisyklės:

$$(\Box) \frac{F}{\Box F}$$

$$(MP) \frac{F \quad F \rightarrow G}{F}$$

Samprotaujant apie troškimus vietoje apibrėžime pateikiamo modalinio operatoriaus $\Box$ naudojamas modalinis operatorius **D**, o samprotaujant apie ketinimus – modalinis operatorius **I**.

3. BDI logika

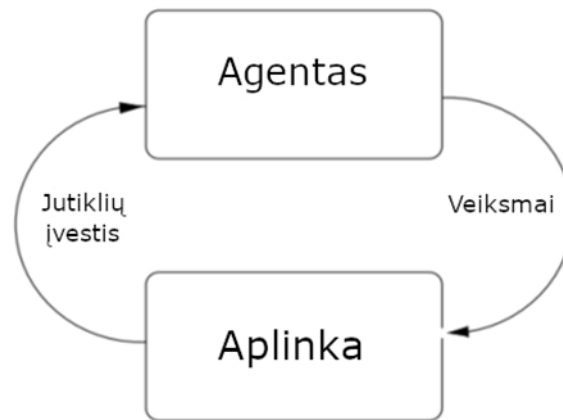
Šiame skyriuje aprašoma BDI logika. Iš pradžių pateikiamas agento apibrėžimas, toliau paaiškinama kas yra BDI modelis. Galiausiai viskas apjungiamą į BDI logikos apibrėžimą.

3.1. Agentai

Agentai yra esybės, operuojančios tam tikroje aplinkoje ir su ja sąveikaujančios.

Kaip nurodoma [Woo00], agentai pasižymi šiomis esminėmis savybėmis:

1. Agentai yra autonomiški. Jiems nereikia įvesties ir prižiūrinčio žmogaus, nes jų veiklą nulemia juose apibrėžta veikimo logika.
 2. Agentai yra iniciatyvūs. Jie elgiasi taip, kad pasiektų savo tikslus.
 3. Agentai yra prisitaikantys. Jie sugeba reaguoti į besikeičiančią aplinką.
 4. Agentai yra socialūs. Jie geba sąveikauti su kitais aplinkoje esančiais agentais.
- Agento veikimą galima pavaizduoti nesudėtinga schema:



Pav. 2. Abstrakti agento veikimo schema. [Woo00]

Žiūrint abstrakčiai, agentų veikimo principas yra paprastas: agentas priima įvestį iš aplinkos, atsižvelgdamas į įvestį priima sprendimą dėl tolimesnių veiksmų ir atlieka pasirinktus veiksmus (sąveikauja su aplinka). Schemoje nevaizduojama agento vidinė logika. Tokia agento veikimo schema tinka bet kokiems agentams.

3.2. „Belief–Desire–Intention“ modelis

„Belief–Desire–Intention“ (BDI) modelis – tai modelis, kuris apibrėžia ir leidžia modeliuoti agentus kaip esybes, turinčias įsitikinimus, troškimus ir ketinimus. Kaip nurodoma [GPP+99], tai vienas populiariausių modelių protingų agentų kūrimo srityje.

BDI modelyje agentus apibrėžia ir jų veiklą lemia šios charakteristikos:

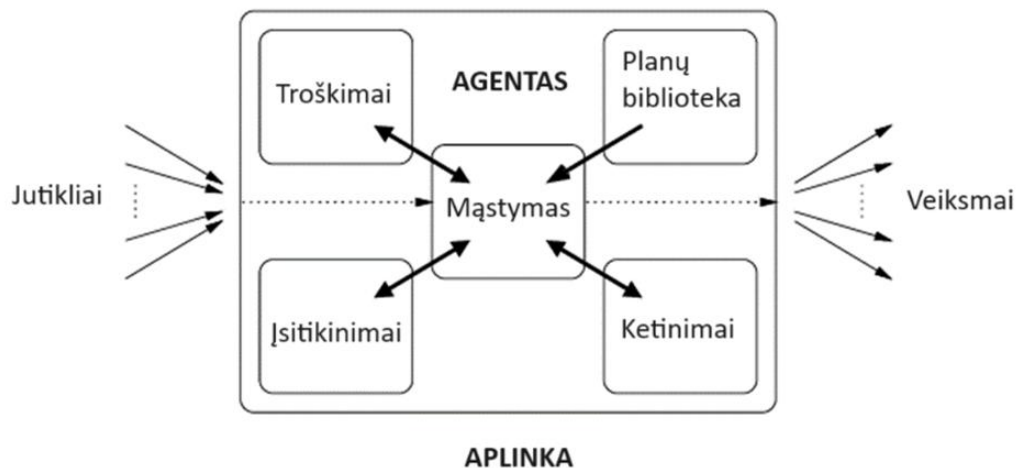
1. Įsitikinimai (angl. **Beliefs**). Tai duomenys apie agento aplinką, kuriuos agentas laiko teisingais.
2. Troškimai (angl. **Desires**). Tai agento esminiai siekiai. Agento veikla ilguoju periodu yra orientuota į šių siekių įgyvendinimą.
3. Ketinimai (angl. **Intentions**). Tai trumpuoju (esamuoju) periodu besireiškiantys (einamieji) agento tikslai.

BDI modeliu paremta žemiau esančiame poskyryje aprašyta BDI logika, kuria remiantis modeliuojami protingi agentai. Šie agentai veikia pagal anksčiau pateiktą agentų veikimo schema, bet jų veiksmų pasirinkimą lemia ne tik aplinka, tačiau ir įsitikinimai, troškimai bei ketinimai.

3.3. Agentų veikimo principai BDI logikoje

Šiame poskyryje aprašoma BDI logika ir ja grįstų agentų veikimo principai.

BDI logika – tai logika modeliuojanti BDI agentų veikimą. Šia logika apibrėžiami agentų įsitikinimai, troškimai ir ketinimai. Šia logika modeliuojamų agentų veikimo procesas yra gan artimas žmogiškajam - jis paremtas įsitikinimais ir iš jų kylančiais ilgalaikiais ir trumpalaikiais tikslais. Žemiau (Pav. 3) pateikiama paprasta BDI agento veikimo schema:



Pav. 3. BDI agento komponentai. [NS04]

Schemoje vaizduojamas BDI agento veikimo principas:

- Agentas priima įvestį iš sensorių.

- Sensorių duomenys patenka į agento „mąstymo“ komponentą.
- Mąstymo komponente atsižvelgiant į esamus įsitikinimus, troškimus, siekius ir planus bei naujausią sensorių įvestį nusprendžiama kuris planas iš planų bibliotekos bus vykdomas, taip pat atnaujinami įsitikinimai, troškimai ir siekiai.

Agento mąstymo procesas pseudokodo forma pateikiamas žemiau [Bir10]:

```

B := B0; /* B0 yra pradiniai agento įsitikinimai */
I := I0; /* I0 yra pradiniai agento ketinimai */
while true do:
    get next percept p;
    B := brf(B, p);
    D := options(B, I);
    I := filter(B, D, I);
    π := plan(B, I);
    execute(π);
end while

```

Čia p – informacija, gauta iš aplinkos, B , D , I – agento įsitikinimų, ketinimų ir troškimų aibės, π – agento vykdymui pasirinktas planas.

Taigi BDI agentas pradeda veikti turėdamas pradinius savo įsitikinimus ir siekius, kurie yra nuolat atnaujinami gaunant informaciją iš aplinkos. Iš pradžių atnaujinami įsitikinimai, tuomet troškimai, o vėliausiai ketinimai. Tuomet iš galimų planų atsižvelgiant į atnaujintus įsitikinimus ir ketinimus pasirenkamas planas, kuris yra įvykdomas.

Agento įsitikinimai, troškimai ir ketinimai apibrėžiami BDI logikos formulėmis. Tuo tarpu plano pasirinkimo procesas remiasi išvedimo metodais, skirtais BDI logikai. Šie metodai gali būti: lentelių metodai, sekvenciniai skaičiavimai, rezoliucijų metodai ir kt. Plano pasirinkimą lemia formulių, susietų su agento veiksmis, išvedamumas ar neišvedamumas, kaip prielaidas naudojant agento įsitikinimus, troškimus ir ketinimus.

Dažniausiai naudojamas formulių išvedimo metodas yra sekvenciniai skaičiavimai. Jie ir yra nagrinėjami šiame darbe. Kaip išvedimo metodą naudojant sekvencinius skaičiavimus planai „svarstomi“ bandant išvesti juos atitinkančias sekvencijas, naudojant automatizuotą skaičiavimą

BDI logikai. Jei planą atitinkančią sekvenciją pavyksta išvesti (įrodyti jos teisingumą) kaip prielaidas naudojant agento įsitikinimus, troškimus ir ketinimus, tuomet planas pasirenkamas vykdymui. [Bir06b]

Akivaizdu, kad BDI agento veikimo procese yra itin svarbus automatizuotas skaičiavimas. Kuo jis efektyvesnis, tuo greičiau ir sklandžiau veiks agentas. Tuo tarpu skaičiavimas, nepasižymintis baigtinumu, gali lemti agento amžiną įstrigimą skaičiavimo (neveiksnumo) būsenoje.

BDI agentų sistemų kūrimui egzistuoja speciali „į agentus orientuotą“ (angl. “agent-oriented”) BDI logikos modeliu pagrįsta programavimo kalba Jason. Jason kaip pagrindą naudoja kitą „į agentus orientuotą“ programavimo kalbą AgentSpeak. Jason kūrėjų tikslas buvo remiantis AgentSpeak sukurti praktiškai pritaikomą įrankį multi-agentų sistemų kūrimui [BH06].

AgentSpeak kalboje agentas apibrėžiamas priskiriant jam pradinį įsitikinimą, sudarytus iš predikatų logikos atominių formulių, ir planų aibę (biblioteką). Šioje kalboje egzistuoja dviejų tipų planai: siektini planai (žymimi !) ir tikrintini planai (žymimi ?). Agentas siekia įgyvendinti siektinus planus. Tuo tarpu tikrintinas planas ?F reiškia, kad agentas turi patikrinti ar F priklauso jo įsitikinimų aibei. Planai apibrėžiami nurodant:

1. Į kokį pasikeitimą - pridėjimą ar ištrynimą - įsitikinimų ar planų reaguodams agentas turėtų juos svarstyti (žymima + arba -).
2. Kontekstą - konjunkciją įsitikinimų, kurie turėtų būti teisingi, kad planas būtų vykdomas,
3. Veiksmų (potikslų) seką, kuri turėtų būti vykdoma, vykdant planą.

Žemiau (Pav. 4.) pateikiamas trijų planų pavyzdys (trečio plano pateikiama tik dalis):

```

+green_patch(Rock)
  : not battery_charge(low)
    <- ?location(Rock,Coordinates);
      !traverse(Coordinates);
      !examine(Rock).

+!traverse(Coords)
  : safe_path(Coords)
    <- move_towards(Coords).

+!traverse(Coords) :
  : not safe_path(Coords)
    <- ...

```

Pav. 4. Pavyzdiniai agento planai AgentSpeak kalboje. [BH06]

Paveikslėlyje matomi planai nurodo kaip agentui (marsaeigiui) reikėtų elgtis pastebėjus žalią plotą ant uolos: jei baterija nėra per daug išsekusi, agentas turėtų surasti akmens lokaciją savo žinių bazėje, tuomet nusigauti iki akmens ir jį apžiūrėti. Nusigavimas iki akmens (traverse) turi dvi matomas alternatyvas: viena iš jų naudojama, kai agentas žino saugų kelią iki akmens, kita, kai saugaus kelio agentas nežino.

4. Sekvenciniai skaičiavimai modalinėms logikoms

Šiame skyriuje apžvelgiama ciklų susidarymo problema sekvenciniuose skaičiavimuose modalinėms logikoms. Vėliau apibrėžiami sekvenciniai skaičiavimai BDI logikai su tiesine laiko logika. Visų pirma apibrėžiamas žinomas skaičiavimas su silpninimo taisykle, šiame darbe vadinamas SBDI*. Toliau apibrėžiamas skaičiavimas SBDI be silpninimo taisyklės ir su procedūra, skirta patikrinti sekvencijos išvedamumą skaičiavime SBDI.

4.1. Ciklai sekvenciniuose skaičiavimuose modalinėms logikoms

Konstruojant sekvencijos išvedimo paieškos medį gali atsirasti tam tikrų pasikartojimų. Jei po taisyklės pritaikymo gaunama sekvencija, kuri jau buvo gauta anksčiau, toliau taikant taisyklės tokia pat eilės tvarka (pagal procedūrą) gauta sekvencija neišvengiamai kartosis. Taigi, medyje atsiranda ciklas. Formalus ciklo apibrėžimas kartu su keliais kitais reikalingais apibrėžimais pateikiamas žemiau.

Apibrėžimas 9. S_0 yra sekvencijos S_{k+1} **protėvis**, jei sekvencijos išvedimo paieškos medyje egzistuoja sekvencijų seka $S_0, S_1, \dots, S_k, S_{k+1}$, kurioje kiekvienam $i = 0, \dots, k$ S_{i+1}, S_i yra atitinkamai prielaida ir išvada kurios nors sekvencinio skaičiavimo taisyklės pritaikymo. [Bir10]

Apibrėžimas 10. Sakoma, kad sekvencijos išvedimo paieškos medyje egzistuoja **griežtas ciklas** $S_0 \rightarrow S'$, jei:

1. Sekvencija S_0 yra sekvencijos S' protėvis.
2. $a(S_0) = a(S')$, $s(S_0) = s(S')$, kur $a(S)$ – sekvencijos S antecedentas, $s(S)$ – sekvencijos S sukcedentas.

Apibrėžimas 11. Sakoma, kad sekvencijos išvedimo paieškos medyje egzistuoja **ciklas** $S_0 \rightarrow S'$, jei:

1. Sekvencija S_0 yra sekvencijos S' protėvis.
2. $a(S_0) \subseteq a(S')$, $s(S_0) \subseteq s(S')$, kur $a(S)$ – sekvencijos S antecedentas, $s(S)$ – sekvencijos S sukcedentas.

Žemiau pateikiamas ciklinio sekvencinio skaičiavimo tiesinei laiko logikai su cikline aksioma apibrėžimas iš [AP13].

Apibrėžimas 12. Ciklinis sekvencinis skaičiavimas tiesinei laiko logikai yra skaičiavimas, kuriame yra aksioma ir taisyklės loginiams operatoriams, apibrėžtos 1.2 poskyryje, ciklinė aksioma (jos apibrėžimas pateikiamas žemiau) bei taisyklės laiko operatoriams:

$$\begin{aligned}
 (\circ) \quad & \frac{\Gamma \vdash \Delta}{\circ \Gamma \vdash \circ \Delta} \\
 (\Box \vdash) \quad & \frac{F, \circ \Box F, \Gamma \vdash \Delta}{\Box F, \Gamma \vdash \Delta} \\
 (\vdash \Box) \quad & \frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, \circ \Box F}{\Gamma \vdash \Delta, \Box F}
 \end{aligned}$$

Apibrėžimas 13. Sekvencija S' yra **ciklinė aksioma**, jeigu egzistuoja sekvencija S , tokia, kad:

1. Egzistuoja ciklas $S \rightarrow S'$ išvedimo paieškos medyje.
2. Cikle tarp S ir S' yra taisyklės $(\vdash \Box)$ taikymo dešinioji prielaida ir nėra taisyklės $(\vdash \Box)$ taikymo kairiosios prielaidos. [AP13]

Pastaba: ciklinės aksiomos apibrėžime naudojama ciklo, o ne griežto ciklo sąvoka, nes ciklinei aksiomai identifikuoti užtenka nustatyti, kad susidarė ciklas, kuris nebūtinai yra griežtas ciklas. Tuo tarpu sekvencijos neišvedamumas nustatomas tik identifikavus griežtą ciklą, kadangi papildomos formulės gali lemti sekvencijos išvedamumą. Toliau šiame darbe apibrėžimuose, teiginiuose ir įrodymuose, nemažinant bendrumo, naudojama ciklo sąvoka.

Jei vykdant skaičiavimus išvedimo paieškos medyje atsiradus sekvencijai S aptinkamas ciklas, šakoje, kurioje yra ši sekvencija, toliau skaičiavimų tęsti nebereikia, belieka patikrinti ar tai blogas ciklas (kuris lemia, kad išvedimo medžio šakoje nebus išvesta aksioma), ar ciklinė aksioma. Todėl vykdant skaičiavimą svarbu gebėti aptikti ciklus.

Paminėtina, kad yra žinomos įvairios strategijos ciklą aptikimo efektyvumui didinti. Pavyzdžiui: operatorių žymėjimas ([Bir06a]), indeksuotos formulės ([Bir08]), sekvencijų žymėjimas ([Bir08]), sekvencijos su istorijomis ([How98]) ir kt.

Žemiau (Pav. 5) pateikiami paprasti griežto ir negriežto ciklo susidarymo sekvencijos išvedimo paieškos medyje pavyzdžiai, iliustruojantys apibrėžimus 10, 11.

$$\begin{array}{c}
 \frac{\dots}{p, \circ\Box p \vdash r} (\vdash \Box) \quad \frac{\frac{S_1^* : \Box p \vdash \Box r}{\circ\Box p \vdash \circ\Box r} (\circ)}{p, \circ\Box p \vdash \circ\Box r} (W \vdash) \\
 \hline
 \frac{p, \circ\Box p \vdash \Box r}{S_1 : \Box p \vdash \Box r} (\vdash \Box) \quad \frac{S_0^* : p, \circ\Box p \vdash \circ\Box\Box r}{\Box p \vdash \circ\Box\Box r} (\Box \vdash) \\
 \hline
 \frac{S_1 : \Box p \vdash \Box r \quad \Box p \vdash \circ\Box\Box r}{\Box p \vdash \Box\Box r} (\vdash \Box) \\
 \hline
 \frac{\Box p \vdash \Box\Box r}{S_0 : \circ\Box p \vdash \circ\Box\Box r} (\circ)
 \end{array}$$

Pav. 5. Griežto ir negriežto ciklų pavyzdžiai sekvencijos išvedimo paieškos medyje.

Pavyzdyje $S_0 \rightarrow S_0^*$ yra negriežtas ciklas, nes S_0^* antecedente yra papildoma formulė p , kurios nėra S_0 . Tuo tarpu $S_1 \rightarrow S_1^*$ yra griežtas ciklas, nes sekvencijų S_1 ir S_1^* antecedentai ir sukcedentai lygūs.

Dažniausiai (griežto) ciklo susidarymas sekvencijos išvedimo paieškos medyje reiškia, kad šaka, kurioje ciklas susidarė, nebus uždaryta, t. y. nesibaigs aksioma. Tačiau išimtis yra aukščiau minėta ciklinė aksioma. Sekvencijos išvedimo paieškos medžio šakoje išvesta ciklinė aksioma reiškia, kad šaka yra išvedama.

Pavyzdyje ciklas $S_0 \rightarrow S_0^*$ yra ciklinė aksioma, nes cikle tarp S_0 ir S_0^* yra $(\vdash \Box)$ taisyklės taikymo dešinioji prielaida ir nėra šios taisyklės taikymo kairiosios prielaidos. Vadinasi šaka, užsibaigianti sekvencija S_0^* , yra išvedama.

4.2. Sekvencinis skaičiavimas BDI logikai SBDI*

Šiame poskyryje aprašomas sekvencinis skaičiavimas BDI logikai su tiesine laiko logika.

Apibrėžimas 14. Sekvencinis skaičiavimas BDI logikai SBDI* - tai skaičiavimas, kuriame be aksiomos ir taisyklių, apibrėžtų 1.2 poskyryje, yra:

- modalinės taisyklės tiesinio laiko modalumui (apibrėžimas 12).
- modalinės taisyklės įsitikinimų (belief), troškimų (desire), ir ketinimų (intention) modalumams:

$$(BEL) \frac{\Gamma, \mathbf{B}\Gamma \vdash G, \mathbf{B}G, \mathbf{B}\Delta}{\mathbf{B}\Gamma \vdash \mathbf{B}G, \mathbf{B}\Delta}$$

$$(DES) \frac{\Gamma \vdash G}{\mathbf{D}\Gamma \vdash \mathbf{D}G}$$

$$(INT) \frac{\Gamma \vdash G}{\mathbf{I}\Gamma \vdash \mathbf{I}G}$$

Pastaba: taisyklėse (BEL) , (DES) , (INT) G – viena formulė, kuri gali būti ir tuščia.

- ciklinė aksioma (apibrėžimas 13).

Pastaba: vietoje 1.2 poskyryje pateiktų taisyklių $(\vdash W)$, $(W \vdash)$ šiame skaičiavime naudojama silpninimo taisyklė (W) , leidžianti nuvalyti bet kokią kiekį formulių nuo sukcedento ir/ar antecedento:

$$(W) \frac{\Gamma \vdash \Delta}{\Gamma', \Gamma, \Gamma'' \vdash \Delta', \Delta, \Delta''}$$

4.3. Sekvencinis skaičiavimas BDI logikai SBDI su procedūra

Šiame poskyryje apibrėžiamas sekvencinis skaičiavimas SBDI su procedūra. Šis skaičiavimas gaunamas modifikavus skaičiavimą SBDI*.

Apibrėžimas 15. Sekvencinis skaičiavimas BDI logikai SBDI - sekvencinis skaičiavimas, kuriame yra aksiomos:

- $\Gamma, F \vdash \Delta, F$
- ciklinė aksioma (apibrėžimas 13)

loginės taisyklės:

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

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

$$(\& \vdash) \frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \& G \vdash^P \Delta}$$

$$(\vdash \&) \frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, G}{\Gamma \vdash^P \Delta, F \& G}$$

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

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

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

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

ir modalinės taisyklės:

$$(WBEL) \frac{\Gamma, \mathbf{B}\Gamma \vdash^P G, \mathbf{B}G, \mathbf{B}\Delta}{\Gamma', \mathbf{B}\Gamma \vdash^P \Delta', \mathbf{B}G, \mathbf{B}\Delta}$$

$$(WDES) \frac{\Gamma \vdash^P G}{\Gamma', \mathbf{D}\Gamma \vdash^P \Delta', \mathbf{D}G, \mathbf{D}\Delta}$$

$$(\Box \vdash) \frac{F, \Box F, \Gamma \vdash \Delta}{\Box F, \Gamma \vdash^P \Delta}$$

$$(W \circ) \frac{\Gamma \vdash^P \Delta}{\Gamma', \circ \Gamma \vdash^P \Delta, \Delta'}$$

$$(WINT) \frac{\Gamma \vdash^P G}{\Gamma', \mathbf{I}\Gamma \vdash^P \Delta', \mathbf{I}G, \mathbf{I}\Delta}$$

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

Čia

- taisyklėje (*WBEL*): Γ', Δ' - formulių aibės, kuriose nėra formulių, suvaržytų operatoriumi **B**,
- taisyklėje (*W* $\circ$): Γ', Δ' - formulių aibės, kuriose nėra formulių, suvaržytų operatoriumi $\circ$,
- taisyklėje (*WDES*): Γ', Δ' - formulių aibės, kuriose nėra formulių, suvaržytų operatoriumi **D**, formulė G – viena iš sukcedente esančių formulių, suvaržytų operatoriumi **D** (arba jokios formulės, kai sukcedente nėra formulių suvaržytų operatoriumi **D**), Δ – formulių, suvaržytų operatoriumi **D**, aibė,
- taisyklėje (*WINT*): Γ', Δ' - formulių aibės, kuriose nėra formulių, suvaržytų operatoriumi **I**, formulė G – viena iš sukcedente esančių formulių, suvaržytų operatoriumi **I** (arba jokios formulės, kai sukcedente nėra formulių suvaržytų su **I**), Δ – formulių, suvaržytų operatoriumi **I**, aibė.
- $P \in \{p, \emptyset\}$. Sekvencijos žymėjimas $\vdash^P$ reiškia, kad pažymėta sekvencija yra gauta pritaikius modalinę taisyklę su integruotu silpninimu, gali būti ciklo pradžios ar pabaigos sekvencija ir yra tikrintina ciklų paieškos metu.

Pastaba: skaičiavime SBDI sekvencijų antecedentai ir sukcedentai yra formulių aibės, ne sekos, todėl formulės juose negali kartotis.

Sekvencinis skaičiavimas SBDI gaunamas iš skaičiavimo SBDI* pašalinus taisykles (*W*), (*C* $\vdash$), ($\vdash C$), (*P* $\vdash$), ($\vdash P$) ir taisykles (*BEL*), (*DES*), (*INT*), ($\circ$) pakeitus modalinėmis taisyklėmis su integruotu silpninimu (*WBEL*), (*WDES*), (*WINT*), (*W* $\circ$).

Skačiavime SBDI silpninimo taisyklės integravimas į modalinės logikos taisykles atliekamas tiksliai, siekiant supaprastinti sekvencijos išvedamumo šiame skaičiavime patikrinimą.

4.3.1. Procedūra skaičiavimui SBDI

Šiame skirsnyje apibrėžiama baigtinė procedūra, patikrinanti ar duota sekvencija S išvedama skaičiavime SBDI.

Norint apibrėžti tokią procedūrą, patogu pirmiausia apibrėžti primarinę sekvenciją ir taisyklių apverčiamumą.

Apibrėžimas 16. Sakome, kad taisyklė **apverčiama**, jei ji pasižymi tokia savybe: jei taisyklės taikymo išvada išvedama, tai ir visos taisyklės prielaidos yra išvedamos [Bir10].

Apibrėžimas 17. Primarinė sekvencija – tai sekvencija:

$$- \quad \Pi, \circ\Gamma_1, \mathbf{B}\Gamma_2, \mathbf{I}\Gamma_3, \mathbf{D}\Gamma_4 \vdash \Sigma, \circ\Delta_1, \mathbf{B}\Delta_2, \mathbf{I}\Delta_3, \mathbf{D}\Delta_4$$

Čia Π, Σ - formulių aibės iš atominių formulių, $\bullet\Gamma_i, \bullet\Delta_i, i = 1, 2, 3, 4$,

$\bullet \in \{\circ, \mathbf{B}, \mathbf{I}, \mathbf{D}\}$ – formulių aibės, kuriose visos formulės suvaržytos operatoriumi $\bullet$ ir kuriose gali būti ir ne atominių formulių.

Pastebėtina, kad primarinei sekvencijai SBDI skaičiavime negalima taikyti jokių taisyklių, išskyrus ($WBEL$), ($WDES$), ($WINT$), ($W \circ$).

Žemiau apibrėžiama procedūra skaičiavimui SBDI yra reikalinga dėl ciklų susidarymo galimybės. Sekvenciniuose skaičiavimuose, kuriuose negali susidaryti ciklai, sekvencijos išvedimo (patikrinimo) procedūra nėra būtina, nes tiesiog taikant skaičiavime apibrėžtas taisyklės galų gale arba gaunama, kad pradinė sekvencija išvedama, arba neišvedus pradinės sekvencijos išnaudojami visi galimi taisyklių taikymai (tokiu atveju pradinė sekvencija neišvedama). Kadangi nagrinėjamame skaičiavime BDI logikai gali susidaryti ciklai ir šio darbo tikslas yra pagerinti skaičiavimo efektyvumą ciklų paieškos atžvilgiu, apibrėžiama procedūra, kurioje nurodoma kada ir kaip bus vykdoma ciklų paieška, kokia eilės tvarka bus taikomos taisyklės.

Apibrėžimas 18. Procedūra P sekvencijos S išvedamumui skaičiavime SBDI patikrinti:

1. Jei S – aksioma, ji išvedama. Procedūra grąžina 1.
2. Jei įmanoma, pasirenkamas loginės taisyklės ar taisyklės laiko modalumui ($\Box \vdash$), ($\vdash \Box$) taikymas, kurį galima pritaikyti sekvencijai S . Gautoms sekvencijoms S_1 ir S_2 (jos gali ir nebūti), kviečiama procedūra (einama į 1-ą žingsnį). Jei gaunama, kad S_1 arba S_2 (jeigu egzistuoja), neišvedama, procedūra grąžina 0.

3. Jei negalimas taisyklės taikymas 2-ame žingsnyje, gauta primarinė sekvencija. Jai nustatomos visos taisyklių $(W \circ)$, $(WBEL)$, $(WDES)$, $(WINT)$ taikymo galimybės. Suformuojama taikymų eilė Q , taikymus į ją dedant tokia tvarka: $(W \circ)$, $(WBEL)$, $(WDES)$, $(WINT)$. Skirtingi tos pačios taisyklės taikymai į eilę sudedami pasirinktinai.
4. Jei eilė Q netuščia, iš eilės imamas taisyklės T taikymas, jį pritaikius sekvencijai S_p gaunama sekvencija S_p' (pritaikius bet kokią taisyklę iš eilės gaunama viena prielaida). Patikrinama, ar išvedus S_p' susidarė ciklas - S_p' palyginama su visais savo protėviais, pažymėtais $\vdash^P$ žymėjimu:
 - a. Jei išvedus sekvenciją S_p' aptikta ciklinė aksioma – grąžinamas 1.
 - b. Jei aptinkamas ciklas, kuris nėra ciklinė aksioma, bandomas kitas taikymas iš Q - grįžtama į 4-ą žingsnį.
 - c. Jei ciklas neaptinkamas, sekvencijai S_p' kviečiama procedūra P (keliaujama į 1-ą žingsnį). Jei S_p' – išvedama, grąžinamas 1.
5. Jei eilėje Q nebelieka neišbandytų taikymų ir nei viena sekvencija, gauta taikant galimus taikymus, nėra išvedama, procedūra grąžina 0.

Žemiau pateikiamas procedūros P pseudokodas „Python“ programavimo kalba.

```

# funkcija, patikrinanti ar sekvencija S išvedama skaičiavime SBDI
def is_derivable_sbdi(S: Sequent) -> bool:
    if is_axiom(S):
        return True

    # jei galima, pritaikoma apverčiama taisyklė
    rule_application = choose_invertible_rule(S)
    if rule_application is not None:
        S1, S2 = apply(S, rule_application)
        s1_derivable = is_derivable_sbdi(S1)
        s2_derivable = is_derivable_sbdi(S2) if S2 is not None else True
        # sekvencija išvedama, jei visos šakos užsidarė
        return s1_derivable and s2_derivable

    # gauta primarinė sekvencija, todėl formuojama neapverčiamų taisyklių eilė
    modal_queue = form_modal_rule_application_queue(S)
    while not modal_queue.empty():
        modal_rule_application = modal_queue.get()
        # modalinės taisyklės iš modalinių taisyklių eilės turi vieną prielaidą
        S1, _ = apply(S, modal_rule_application)

        loop_axiom, loop = check_loop(S1)
        if loop_axiom:
            return True

        if loop:
            continue

        s1_derivable = is_derivable_sbdi(S1)

        # jei pavyko išvesti sekvenciją S1, tai S išvedama
        if s1_derivable:
            return True

    # Jei po visų taikymų gavome neišvedamas sekvencijas, S neišvedama
    return False

```

Pav. 6. Procedūros P pseudokodas „Python“ programavimo kalba.

5. Skaičiavimo SBDI savybės

5.1. Skaičiavimų SBDI* ir SBDI ekvivalentumas

Teiginys 1: Naudojant procedūrą P skaičiavimui SBDI išvedamos tos ir tik tos sekvencijos, kurios išvedamos skaičiavime SBDI*.

Teiginio 1 įrodymas:

- Žymėjimai, naudojami įrodyme:
 - S – sekvencija, kuri buvo išvesta skaičiavime SBDI*.
 - M^* – sekvencijos S išvedimo medis skaičiavime SBDI*.
 - M – sekvencijos S išvedimo medis skaičiavime SBDI su procedūra P.
 - $A = \{(WBEL), (WDES), (WINT), (W \circ)\}$ – modalinių taisyklių, su integruota taisykle (W) aibė skaičiavime SBDI.
 - $A^* = \{(BEL), (DES), (INT), (\circ)\}$ – modalinių taisyklių, atitinkančių taisyklės iš A, aibė skaičiavime SBDI*.
 - S^* – einamoji sekvencija.
 - E_A, E_S – aibės formulių, kurios nuvalomos iš antecedento ir sukcedento (atitinkamai) taikant taisyklės iš aibės A.
 - T^* – taisyklė, taikoma einamajai sekvencijai S^* medyje M^* .
 - S_1, S_2 – sekvencijos, gautos pritaikius taisyklės T^* atitikmenį einamajai sekvencijai S^* . (S_2 gali ir nebūti, priklausomai nuo T^*).
 - PM^* – nuoroda į taisyklės T^* taikymą sekvencijai S^* medyje M^* .
 - PM – nuoroda į vietą medyje M, kurioje bus taikoma taisyklė T^* ar jos atitikmuo sekvencijai S^* .
 - T^{*^*} – kita taisyklė, taikoma atitinkamoje šakoje medyje M^* .
 - PM^{*^*} – nuoroda į taisyklės T^{*^*} taikymo vietą medyje M^* .
 - PM^* – nuoroda į vietą medyje M, kur bus taikoma taisyklė T^{*^*} ar jos atitikmuo.

I.[SBDI* $\rightarrow$ SBDI] Įrodysime, kad jei sekvencija išvedama skaičiavime SBDI*, tai ji išvedama ir skaičiavime SBDI.

- Tegul skaičiavime SBDI* buvo išvesta sekvencija S, t. y. buvo sukonstruotas sekvencijos S išvedimo medis M*. Iš M* galima gauti sekvencijos S išvedimo medį skaičiavime SBDI su procedūra P (medį M) tokiu būdu:
 - Pradžioje $M = \emptyset$ (iš pradžių medis tuščias).
 - Pradinė būseną – $B_0 = \langle S^{\setminus}, S, E_A = \emptyset, E_S = \emptyset, T^* = T_n, PM^* = PM_0^*, PM = PM_0 \rangle$:
 - T_n - taisyklė kuri yra taikoma pradinei sekvencijai S medyje M*.
 - PM_0^* - nuoroda į T_n taikymą medyje M*.
 - PM_0 – nuoroda į medžio M šaknį.
 - Einant per medį M* formuojama FIFO (first-in-first-out) būsenų eilė Q, kurioje pradžioje yra tik pradinė būseną B_0 .
 - Einama per medį M* nuo S į viršų (taip, kaip taikomos taisyklės) ir iš kairės į dešinę (jei medžio šaka išsišakoja).
 - Paeiliui iš eilės Q imamos būsenos.
 - Atsidūrus būsenoje $B_i = \langle S^{\setminus}, E_A, E_S, T^*, PM^*, PM \rangle$:
 - Jei $S^{\setminus}$ – aksioma (įskaitant ir ciklinę aksiomą), tai ji įrašoma medyje M ir šaka, į kurią rodo PM, užsidaro. Eilė Q nėra papildoma jokia nauja būseną.
 - Jei $T^* = (W)$, tai medyje M ji tiesiog praleidžiama (medis M nepapildomas), bet įsimenama kokios formulės buvo nuvalytos iš $S^{\setminus}$ antecedento ($\{F_1, \dots, F_n\}$) ir sukcedento ($\{G_1, \dots, G_m\}$). Į eilę pridedama būseną $B_j = \langle S^{\setminus}, E_A \cup \{F_1, \dots, F_n\}, E_S \cup \{G_1, \dots, G_m\}, T^{\setminus}, PM^{\setminus}, PM^{\setminus} \rangle$.
 - Jei $T^* \in A^*$, tai vietoje T^* medyje M sekvencijai $S^{\setminus}$ taikoma atitinkama taisyklė $T \in A$. Į M įrašoma sekvencija $S^{\setminus}$ ir taisyklės T taikymas. Į eilę Q pridedama būseną $B_j = \langle S_1, \emptyset, \emptyset, T^{\setminus}, PM^{\setminus}, PM^{\setminus} \rangle$. Taikant taisyklę T sekvencijai $S^{\setminus}$ pirmiausia iš $S^{\setminus}$ antecedento pašalinamos formulės iš E_A , o iš sukcedento – formulės iš E_S .
 - Visais kitais atvejais medyje M taisyklė T^* taikoma sekvencijai $S^{\setminus}$ tokia, kokia buvo medyje M*. Į medį M įrašoma sekvencija $S^{\setminus}$ ir taisyklės T^* taikymas. Į eilę Q pridedama būseną $B_j = \langle S_1, E_A, E_S, T^{\setminus}, PM^{\setminus}, PM^{\setminus} \rangle$ (jei taisyklė T^* turi 2 prielaidas, į eilę taip pat pridedama atitinkama būseną antrajai prielaidai S_2).

Pastaba: taisyklės medyje M taikomos toms pačioms formulėms, kurioms buvo taikomos medyje M^* .

Nuorodos į medį M^* (PM^* , $PM^{*`}$ ) saugomos tam, kad būtų žinoma kokia kita taisyklė taikoma medyje  $M^*$ , t. y. kad būtų galima gauti tinkamą  $T^{*`}$ reikšmę. Jei kitos taikomos taisyklės nėra, nes nagrinėjamos būsenos sekvencija jau yra aksioma ir medyje M^* šaka užsibaigia, tai $T^{*`}$ ,  $PM^{*`}$ – tuščios (null).

- Laikantis aukščiau pateiktų instrukcijų gautas medis M yra sekvencijos S išvedimo skaičiavime SBDI medis, nes:
 - Bet koks taisyklės taikymas iš M^* turi savo atitikmenį M .
 - Medžio M šakos, kuriose nėra silpninimo taisyklės ar taisyklių iš A taikymo, yra identiškos atitinkamoms medžio M^* šakoms, įskaitant ir aksiomas.
 - Medžio M šakos, kuriose yra silpninimo taisyklės taikymas ir nėra taisyklių iš A taikymo, skiriasi tuo, kad medyje M^* nuvalytos formulės lieka medyje M ir medžio M šakoje praleistos silpninimo taisyklės. Taigi medyje M visos sekvencijos virš silpninimo taisyklės taikymų, lyginant su M^* , turi papildomų formulių, įskaitant ir aksiomas.
 - Medžio M šakos, kuriose yra taisyklės iš A taikymas ir nėra silpninimo taisyklės taikymo, nuo atitinkamų M^* šakų skiriasi tik tuo, kad taisyklės iš A^* yra pakeistos atitinkamomis taisyklėmis iš A .
 - Medžio M šakos, kuriose yra silpninimo taisyklės taikymas ir taisyklių iš A taikymas, skiriasi nuo atitinkamų M^* šakų šitaip:
 - Vietoje taisyklių iš A^* yra atitinkamos taisyklės iš A .
 - Nėra silpninimo taisyklės taikymų.
 - Formulės, kurios buvo nuvalytos su silpninimo taisyklėmis, nuvalomos taikant taisykles iš A , t. y. vėliau nei medyje M^* , nes ten šios formulės buvo nuvalytos **prieš** taikant taisyklę iš A^* .
 - Formulės, kurios buvo nuvalytos po paskutinio taisyklės iš A^* taikymo, lieka medyje M .
 - Jokie galimi medžių M^* ir M šakų skirtumai nepanaikina aksiomų, kuriomis užsibaigia medžio M^* šakos - medis M užsibaigia tokiomis pačiomis aksiomomis arba aksiomomis, papildytais formulėmis, kurios buvo nuvalytos medyje M^* .

- Belieka parodyti, kad skaičiavimas SBDI išlieka pilnas laikantis procedūros P:
 1. Tegul S – kokia nors skaičiavime SBDI išvedama sekvencija.
 2. Konstruojant S išvedimo medį laikantis procedūros P, kol galima, bus taikomos apverčiamos taisyklės ir visos naujai gautos sekvencijos bus išvedamos skaičiavime SBDI. Jei pakeliui bus gauta aksioma, šaka bus uždaryta.
 3. Pirmą kartą išvedus primarinę sekvenciją S_p ji tikrai bus išvedama skaičiavime SBDI, nes S išvedama ir buvo taikytos tik apverčiamos taisyklės. Vadinasi kuri nors sekvencija S^* gauta atlikus vieną iš galimų neapverčiamų taisyklių taikymų tikrai bus išvedama skaičiavime SBDI. Kadangi bus perrenkami visi neapverčiamų taisyklių taikymai, tai kažkada būtinai bus gauta išvedama sekvencija S^* . Jei S^* aksioma, šaka bus uždaryta.
 4. Toliau induktyviai taikant žingsnius (2), (3) naujai gautai išvedamai sekvencijai S^* , pradinė sekvencija S tikrai bus išvesta, nes procedūroje P numatytas ciklų aptikimas, taigi jokios neišvedamos sekvencijos nebus bandoma išvesti amžinai, ir galų gale bus rasta viena iš taisyklių taikymų sekų, sukonstruojanti pradinės sekvencijos išvedimo medį.
 5. Vadinasi laikantis procedūros P skaičiavimui SBDI yra išvedamos tos ir tik tos sekvencijos, kurios yra išvedamos skaičiavime SBDI be procedūros P (procedūros laikymasis negali praplėsti išvedamų sekvencijų aibės).

II. $[SBDI \rightarrow SBDI^*]$ Įrodysime, kad jei sekvencija išvedama skaičiavime SBDI, tai ji išvedama ir skaičiavime SBDI*.

- Tegul skaičiavime SBDI su procedūra P buvo išvesta sekvencija S , t. y. buvo sukonstruotas sekvencijos S išvedimo medis M .
 - Iš M galima gauti sekvencijos S išvedimo medį skaičiavime SBDI* (medį M^*) analogiškai pradedant nuo pradinės sekvencijos S ir einant per medyje M esančius taisyklių taikymus. Visos taisyklės iš medžio M medyje M^* taikomos taip pat, išskyrus taisykles iš A . $\forall T \in A$ pakeičiama viena arba dviem taisyklėmis:
 1. Taisykle (W), kuri nuvalo tas ir tik tas formules, kurias nuvalė T (jei T nenuvalė jokių formulių, taisyklė (W) netaikoma).
 2. Atitinkama taisykle $T^* \in A^*$:

$$\frac{\Gamma \vdash^p \Delta}{\Gamma', \circ\Gamma \vdash \circ\Delta, \Delta'} (W\circ) \rightarrow \frac{\frac{\Gamma \vdash \Delta}{\circ\Gamma \vdash \circ\Delta} (\circ)}{\Gamma', \circ\Gamma \vdash \circ\Delta, \Delta'} (W)$$

$$\frac{\Gamma \vdash^p G}{\Gamma', \Pi\Gamma \vdash \Delta', \text{IG}, \text{I}\Delta} (WINT) \rightarrow \frac{\frac{\Gamma \vdash G}{\Pi\Gamma \vdash \text{IG}} (INT)}{\Gamma', \Pi\Gamma \vdash \Delta', \text{IG}, \text{I}\Delta} (W)$$

$$\frac{\Gamma, \text{B}\Gamma \vdash^p G, \text{BG}, \text{B}\Delta}{\Gamma', \text{B}\Gamma \vdash \Delta', \text{BG}, \text{B}\Delta} (WBEL) \rightarrow \frac{\frac{\Gamma, \text{B}\Gamma \vdash G, \text{BG}, \text{B}\Delta}{\text{B}\Gamma \vdash \text{BG}, \text{B}\Delta} (BEL)}{\Gamma', \text{B}\Gamma \vdash \Delta', \text{BG}, \text{B}\Delta} (W)$$

$$\frac{\Gamma \vdash^p G}{\Gamma', \text{D}\Gamma \vdash \Delta', \text{DG}, \text{D}\Delta} (WDES) \rightarrow \frac{\frac{\Gamma \vdash G}{\text{D}\Gamma \vdash \text{DG}} (DES)}{\Gamma', \text{D}\Gamma \vdash \Delta', \text{DG}, \text{D}\Delta} (W)$$

Pav. 7. Taisyklių iš SBDI taikymų pakeitimo taisyklių iš SBDI* taikymais schema.

Taigi iš bet kuriame skaičiavime gauto sekvencijos S išvedimo medžio galima sukonstruoti sekvencijos S išvedimo medį kitame skaičiavime. QED.

5.2. Skaičiavimo SBDI baigtinumas

Apibrėžimas 19. Sekvenciniame skaičiavime **and-taisyklė** yra tokia taisyklė, kurios išvada išvedama, jei išvedamos visos prielaidos. [Bir10]

Apibrėžimas 20. Sekvenciniame skaičiavime **or-taisyklė** yra tokia taisyklė, kurios išvada išvedama, jei išvedama bent viena iš prielaidų. [Bir10]

Įprastos sekvencinio skaičiavimo taisyklės reiškia, kad išvada išvedama, jei visos taisyklės taikymo prielaidos yra išvedamos. [Bir10] tokios įprastos taisyklės vadinamos and-taisyklėmis, o šalia jų apibrėžiamos ir naudojamos or-taisyklės. Šios or-taisyklės, kaip nurodoma aukščiau pateiktame apibrėžime 20, reiškia, kad išvadose esanti sekvencija išvedama, jei išvedama bent viena iš prielaidose esančių sekvencijų.

Kalbant apie skaičiavimo baigtinumą svarbus ir anksčiau apibrėžtas apverčiamumas. Apverčiamos taisyklės yra labai patogios – taikant vien jas bet kurioje išvedimo paieškos medžio šakoje gavus neišvedamą sekvenciją iš karto yra žinoma, kad ir pradinė sekvencija yra neišvedama. Yra žinoma, kad loginės taisyklės, apibrėžtos 1.2 poskyryje, yra apverčiamos. Taip pat apverčiamos ir taisyklės tiesinei laiko logikai ($\vdash \Box$), ($\Box \vdash$), ($\circ$). [AP13].

Taikant taisyklę, kuri nėra apverčiama, ir gavus neišvedamą sekvenciją, reikia grįžti atgal ir bandyti vietoj neapverčiamos taisyklės taikymo taikyti kitą, kol bus išnaudotos visos galimybės. Taikant neapverčiamas taisykles atsitiktinai, smarkiai komplikuojasi sekvencijos išvedimo paieškos medžio konstravimo procesas. Norint supaprastinti skaičiavimą reikia apriboti neapverčiamų taisyklių taikymą. Vienas iš būdų tai padaryti yra neapverčiamų taisyklių integravimas į or-taisykles, šitaip pažymint visas vietas, kuriose gali tekti ieškoti taisyklių taikymo alternatyvų.

Skaiciavimas SBDI kartu su procedūra P išsprendžia silpninimo neapverčiamumo problemą kitokiu, bet iš esmės ekvivalenčiu būdu - silpninimo taisyklė integruojama į modalines taisykles, turinčias apribojimų sekvencijos struktūrai, o šios modalinės taisyklės taikomos tik tada, kai apverčiamų taisyklių taikyti negalima. Taigi SBDI su procedūra P užtikrina, kad taisyklės nuvalomos tik tuo atveju, kai tai yra neišvengiama, t. y. kai nenuvalius formulės ar formulių nebegalima taikyti jokių taisyklių.

Procedūra P skaičiavimui SBDI leidžia patikrinti ar sekvencija išvedama skaičiavime SBDI, tikrinant taisyklių taikymo alternatyvas tik tose vietose, kur buvo taikomos taisyklės su integruotu silpninimu. Kadangi bet kokiai baigtinei sekvencijai yra baigtinis kiekis loginių ir modalinių taisyklių taikymo būdų ir į procedūrą P įtrauktas ciklų aptikimas, skaičiavimas SBDI su procedūra P yra baigtinis.

6. Apribojimai ciklams skaičiavime SBDI

Šiame skyriuje suformuluojami ir įrodomi teiginiai apie skaičiavimą SBDI, remiantis šiais teiginiais, suformuluojami apribojimai ciklų susidarymui šiame skaičiavime. Taip pat SBDI pagrindu apibrėžiamas naujas efektyvesnis skaičiavimas su procedūra, į kurią integruojami pateikti apribojimai.

6.1. Apribojimai troškimų ir ketinimų ciklams

Apibrėžimas 21. Sekvencijų S_1 ir S_2 **palyginimo operacijos**. Pažymėkime $a(S)$ – sekvencijos S antecedentas, $s(S)$ – sekvencijos S sukcedentas.

1. $S_1 = S_2$, jei $s(S_1) = s(S_2)$ ir $a(S_1) = a(S_2)$.
2. $S_1 < S_2$, jei $(s(S_1) \subseteq s(S_2) \text{ ir } a(S_1) \subset a(S_2))$ arba $(s(S_1) \subset s(S_2) \text{ ir } a(S_1) \subseteq a(S_2))$.
3. $S_1 > S_2$, jei $S_2 < S_1$.
4. $S_1 \geq S_2$, jei $S_1 = S_2$ arba $S_1 > S_2$.
5. $S_1 \leq S_2$, jei $S_1 = S_2$ arba $S_1 < S_2$.

Teiginys 2: Jei $S_0 \rightarrow S^*$ yra ciklas skaičiavimo SBDI kokios nors sekvencijos išvedimo paieškos medyje, tai tarp S_0 ir S^* nėra taisyklių (*WDES*), (*WINT*) taikymų.

Prieš įrodant **teiginį 2** reikia įvesti formulės ir sekvencijos gylio operatorių atžvilgiu apibrėžimus. Jie pateikiami žemiau.

Apibrėžimas 22. **D**-gylis G_D :

1. $G_D(p) = 0$, jei p – loginis kintamasis.
2. $G_D(\bullet F) = G_D(F)$, jei F – formulė, $\bullet \in \{\neg, \mathbf{B}, \mathbf{I}, \circ, \square\}$.
3. $G_D(F_1 \bullet F_2) = \max(G_D(F_1), G_D(F_2))$, jei F – formulė, $\bullet \in \{\rightarrow, \&, \vee\}$.
4. $G_D(\mathbf{D}F) = 1 + G_D(F)$, jei F – formulė.
5. $G_D(S) = \max(G_D(F), F \in S)$, jei S – sekvencija (sekvencijos **D**-gylis lygus maksimaliam sekvencijos formulės **D**-gyliui).

Apibrėžimas 23. **I**-gylis G_I . **I**-gylis apibrėžiamas analogiškai kaip **D**-gylis, tik operatoriaus **I** atžvilgiu.

Teiginio 2 įrodymas:

1. Tarkime, skaičiavime SBDI kokios nors sekvencijos išvedimo paieškos medyje susidarė ciklas $S_0 \rightarrow S^*$, kur $S^* \geq S_0$ ir tarp sekvencijų S_0 ir S^* buvo taikoma taisyklė (*WDES*).
2. Tuomet turime tokią situaciją:

$$\begin{array}{c}
 \frac{S^*(S^* \succeq S_0)}{\dots} (T_l) \\
 \frac{\dots}{S'' : F_1, \dots, F_n \vdash^P G_i} (T_{k+2}) \\
 \hline
 \frac{S' : DF_1, \dots, DF_n, \Gamma, \circ\Gamma_1, B\Gamma_2, \Pi_3 \vdash DG_1, \dots, DG_m, \Delta, \circ\Delta_1, B\Delta_2, I\Delta_3}{\dots} (WDES) \\
 \hline
 \frac{\dots}{S_0} (T_k) \\
 \hline
 S_0 \quad (T_1)
 \end{array}$$

Pav. 8. Ciklo, kurio viduje buvo taikoma taisyklė (*WDES*), schema.

3. Jei sekvencijos išvedimo paieškos medyje turime taisyklės *T* taikymą, kur S_2 yra prielaida, o S_1 – išvada, tai $G_D(S_2) \leq G_D(S_1)$, nes nėra taisyklės, kuri padidintų kokios nors formulės iš S_1 **D**-gylį (nėra taisyklės, kuri pridėtų operatorių **D** prie kokios nors formulės).
4. Jei turime taisyklės (*WDES*) taikymą, kur S'' yra prielaida, o S' – išvada, tai $G_D(S'') \leq G_D(S') - 1 < G_D(S')$, nes taisyklės (*WDES*) taikymas sekvencijai S' nuvalė visus išorinius operatorius **D**.
5. Iš (3), (4), seka: $G_D(S_0) \geq G_D(S') > G_D(S'') \geq G_D(S^*)$, taigi $G_D(S_0) > G_D(S^*)$.
6. Akivaizdu, kad jei $S_1 \geq S_2$, tai $G_D(S_1) \geq G_D(S_2)$, kadangi maksimalaus **D**-gylio formulė iš S_2 priklauso ir S_1 . Todėl, jei galioja: $G_D(S_0) > G_D(S^*)$, tai negali būti teisinga nelygybė: $S^* \geq S_0$.
7. Gavome prieštarą, vadinasi (1) apibrėžta situacija yra negalima. Taigi, jei tarp dviejų sekvencijų išvedimo paieškos medyje yra taisyklės (*WDES*) taikymas, tos sekvencijos nesudaro ciklo.

Taisyklė (*WINT*) yra identiška taisyklei (*WDES*), todėl jai galioja samprotavimas (1) – (7), jame pakeitus operatorių **D** operatoriumi **I** ir **D**-gylį **I**-gyliu. Vadinasi: jei $S_0 \rightarrow S^*$ yra ciklas

skaičiavimo SBDI kokios nors sekvencijos išvedimo paieškos medyje, tai tarp S_0 ir S^* nėra taisyklių (*WDES*), (*WINT*) taikymų. QED.

Išvada 1. Tegul skaičiavime SBDI sekvencijos išvedimo paieškos medyje pritaikius taisyklę T buvo išvesta sekvencija S^* . Norint patikrinti ar sekvencijos išvedimo paieškos medyje egzistuoja sekvencija S_0 , su kuria susidarytų ciklas $S_0 \rightarrow S^*$, užtenka naujai išvestą sekvenciją S^* lyginti su jos protėviais iki pirmo taisyklių (*WDES*), (*WINT*) taikymo keliaujant nuo S^* sekvencijos išvedimo paieškos medžio šakomis žemyn (nuo prielaidų link išvadų).

Pastebėtina, kad dėl skirtingų modalumų sąveikos multimodalinės logikos skaičiavimuose apribojimai ciklų susidarymui gali būti kitokie nei monomodalinės logikos skaičiavimuose. Pavyzdžiui, išvadoje 1 apibrėžtas apribojimas kalba apie taisykles modaliniams operatoriams **D** ir **I**. Toks apribojimas negalėtų egzistuoti skaičiavimuose monomodalinėms laiko ar įsitikinimų (belief) logikoms, tačiau skaičiavime SBDI šis apribojimas reiškia, kad pritaikius šias taisykles negali susidaryti jokie ciklai, net ir tie, kurių susidarymą lemtų taisyklių operatoriams **B**, $\circ$ ar $\square$ taikymas.

Žemiau pateikiami sekvencijų išvedimo pavyzdžiai skaičiavime SBDI su procedūra P.

$$\begin{array}{c}
 \frac{(ax)}{r \vdash r, s} (\vdash \vee) \\
 \frac{r \vdash r, s}{r \vdash^P r \vee s} (\vdash \vee) \\
 \frac{p, q, \mathbf{I}r \vdash s, \mathbf{I}(r \vee s) : P_2}{p, q \wedge \mathbf{I}r \vdash s, \mathbf{I}(r \vee s)} (WINT) \\
 \frac{p, q \wedge \mathbf{I}r \vdash s, \mathbf{I}(r \vee s)}{p, q \wedge \mathbf{I}r \vdash^P s \vee \mathbf{I}(r \vee s)} (\wedge \vdash) \\
 \frac{(ax)}{p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p, \mathbf{D}(s \vee \mathbf{I}(q \vee s))} \\
 \frac{p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p, \mathbf{D}(s \vee \mathbf{I}(q \vee s))}{p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p \vee \mathbf{D}(s \vee \mathbf{I}(q \vee s))} (\vee \vdash) \\
 \frac{p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p, \mathbf{D}(s \vee \mathbf{I}(q \vee s)) : P_1}{S : \mathbf{D}p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p, \mathbf{D}(s \vee \mathbf{I}(q \vee s))} (WDES) \\
 \frac{S : \mathbf{D}p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p, \mathbf{D}(s \vee \mathbf{I}(q \vee s))}{p \vee \mathbf{D}p, \mathbf{D}(q \wedge \mathbf{I}r) \vdash p \vee \mathbf{D}(s \vee \mathbf{I}(q \vee s))} (\vee \vdash)
 \end{array}$$

Pav. 9. Sekvencijos išvedimo medžio pavyzdys skaičiavime SBDI su procedūra P.

Paveikslėlyje vaizduojamame sekvencijos S išvedimo medyje 2 kartus gaunamos primarinės sekvencijos, jos pažymėtos P_1 ir P_2 . Gautoms primarinėms sekvencijoms galimas tik vienas neapverčiamų taisyklių taikymas.

Toliau pateikiamas dar vienas sekvencijos išvedimo skaičiavime SBDI pavyzdys.

Pav. 10. Sekvencijas išvedimo medis skaičiavime SBDI su cikline aksioma.

Konstruojant šį sekvenčijos išvedimo medį gautoms primarinėms sekvencijoms buvo galima taikyti tiek (*WDES*), tiek ($W \circ$) taisykles. Pavaizduotas išvedimo paieškos medis yra toks, koks būtų sukonstruojamas laikantis procedūros P. Vaizduojant išvedimo medį nevaizduojamos šakos,

kuriuose primarinėms sekvencijoms buvo taikomos taisyklės, lėmusios šakos neužsidarymą ir grįžimą atgal.

Pateiktame galutiniame sekvencijos išvedimo medyje 5 kartus taikomos modalinės taisyklės su integruotu silpninimu (gautos sekvencijos pažymėtos $\vdash^P$):

- 3 kartus taikoma taisyklė $(W \circ)$. Ieškant ciklų pritaikius taisyklę gautos sekvencijos lyginamos su visais jų protėviais, pažymėtais $\vdash^P$.
- 2 kartus taikoma taisyklė $(WDES)$. Netaikant apribojimo iš **išvados 1** ieškant ciklų gautos sekvencijas reikėtų lyginti su visais jų protėviais, pažymėtais $\vdash^P$.

Taigi taikant apribojimą iš **išvados 1** būtų sutaupyta sekvencijų palyginimų. Pabrėžtina, kad taikant apribojimus ciklų paieškai sekvencijos išvedimo medis nesikeičia. Todėl efektyvesnė ciklų aptikimo paieška ir mažesnis sekvencijų tarpusavio palyginimų skaičius taip pat reiškia ir efektyvesnį sekvencijos išvedimo (paieškos) medžio konstravimą.

6.2. Laiko ir įsitikinimų ciklai

Šiame poskyryje pateikiami laiko ir įsitikinimų ciklų, t. y. ciklų, kuriuose taikoms taisyklės $(WBEL)$, $(W \circ)$, pavyzdžiai.

Teiginio 2 įrodyme pateiktas samprotavimas (1) – (7) negalioja taisyklėms $(WBEL)$, $(W \circ)$, $(\vdash \Box)$, $(\Box \vdash)$, nes skaičiavime SBDI yra taisyklių, kurios prideda prie formulių išorinius operatorius $\circ$, $\Box$, o taisyklė $(WBEL)$ nenuvalo išorinių operatorių **B** nuo formulių, kurioms yra taikoma.

$$\begin{array}{c}
 \frac{(ax) \quad \frac{p, q \wedge r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash \mathbf{B}(p \wedge s), p}{S : p, q \wedge r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash^P \mathbf{B}(p \wedge s), p \wedge s}}{\frac{\frac{S' : p, q \wedge r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash^P \mathbf{B}(p \wedge s), p \wedge s}{p, q, r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash \mathbf{B}(p \wedge s), s} (WBEL) \quad \frac{p, q, r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash \mathbf{B}(p \wedge s), s}{p, q \wedge r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash \mathbf{B}(p \wedge s), s} (\wedge \vdash)}{p, q \wedge r, \mathbf{B}p, \mathbf{B}(q \wedge r) \vdash \mathbf{B}(p \wedge s), s} (\vdash \wedge)}{\mathbf{B}p, \mathbf{B}(q \wedge r) \vdash \mathbf{B}(p \wedge s)} (WBEL)
 \end{array}$$

Pav. 11. Ciklo, kuriame yra $(WBEL)$ taisyklės taikymas, pavyzdys skaičiavime SBDI.

Pavyzdyje susidaro ciklas $S \rightarrow S'$, nes $S = S'$ ir S yra S' protėvis sekvencijos išvedimo paieškos medyje. Tai blogas ciklas (griežtas ciklas ir ne ciklinė aksioma), nes jame nėra taisyklės $(\vdash \Box)$ taikymo dešinėsios prielaidos, todėl šaka tęsiant skaičiavimą niekad neužsidarytų. Cikle

$S \rightarrow S'$ yra taisyklės (*WBEL*) taikymas, todėl akivaizdu, kad ieškant ciklų skaičiavime SBDI neužtenka tikrinti sekvencijų iki pirmo šios taisyklės taikymo.

Kadangi, kaip įrodo aukščiau esantis ciklo pavyzdys, **išvada 1** netaikytina kitoms modalinėms taisyklėms skaičiavime SBDI, norint apriboti ciklų susidarymo galimybes ar optimizuoti ciklų paiešką kitoms modalinėms taisyklėms, reikia identifikuoti papildomas skaičiavimo SBDI savybes.

Žemiau pateikiamas „mišraus“ ciklo pavyzdys, iliustruojantis galimą ciklų įvairovę skaičiavime SBDI.

$$\begin{array}{c}
 (ax) \\
 \hline
 \frac{p, \Box p, B\Box p \vdash p, B(p \wedge q)}{\Box p, B\Box p \vdash p, B(p \wedge q)} \quad \frac{S' : p, \Box p, B\Box p \vdash q, B(p \wedge q)}{\Box p, B\Box p \vdash q, B(p \wedge q)} \quad (\Box \vdash) \\
 \hline
 \frac{\Box p, B\Box p \vdash p, B(p \wedge q) \quad \Box p, B\Box p \vdash q, B(p \wedge q)}{\Box p, B\Box p \vdash p \wedge q, B(p \wedge q)} \quad (\vdash \wedge) \\
 \hline
 \frac{\Box p, B\Box p \vdash p \wedge q, B(p \wedge q)}{S : p, \Box p, B\Box p \vdash q, B(p \wedge q)} \quad (WBEL)
 \end{array}$$

Pav. 12. Ciklo, kuriame taikomos taisyklės (*WBEL*), ($\Box \vdash$) pavyzdys.

Pavyzdyje susiformuoja ciklas $S \rightarrow S' : S = S'$ ir S yra S' protėvis. $S \rightarrow S'$ nėra ciklinė aksioma, nes tarp S ir S' nėra taisyklės ($\vdash \Box$) taikymo. Pastebėtina, kad primarinei sekvencijai S galima taikyti taisyklės ($W \circ$) ir (*WBEL*), tačiau pritaikius taisyklę ($W \circ$) sekvencija taip pat nebūtų išvesta. Šis pavyzdys įrodo, kad skaičiavime SBDI gali susidaryti ir mišrūs ciklai, t. y. tokie ciklai, kuriuose yra taikomos taisyklės daugiau nei vienam modalumui.

Dar vienas sudėtingesnio ciklo atvejis – ciklas, kuris susidaro kelis kartus taikant tą pačią modalinę taisyklę. Dažnai tokie ciklai, kuriuose n kartų taikoma kuri nors modalinė taisyklė, vadinami n -ilgio ciklais. Žemiau pateikiamas pavyzdys ciklo, kuriame $n = 2$ kartus taikoma modalinė taisyklė (*WBEL*).

$$\begin{array}{c}
 (C) \\
 \hline
 \frac{S' : Bp, p \vdash^P q, Bq, B(r \vee Bq)}{Bp, p \vdash r, Bq, B(r \vee Bq)} \quad (WBEL) \\
 \hline
 \frac{Bp, p \vdash r, Bq, B(r \vee Bq)}{Bp, p \vdash^P Bq, r \vee Bq, B(r \vee Bq)} \quad (\vdash \vee) \\
 \hline
 \frac{Bp, p \vdash^P Bq, r \vee Bq, B(r \vee Bq)}{S : Bp, p \vdash q, Bq, B(r \vee Bq)} \quad (WBEL)
 \end{array}$$

Pav. 13. Ciklas, kuriame 2 kartus taikoma taisyklė (*WBEL*), skaičiavime SBDI.

6.3. Apribojimai įsitikinimų ir laiko ciklams

Šiame poskyryje pateikiami apribojimai laiko ir įsitikinimų ciklams. Tačiau visų pirma pateikiami reikalingi apibrėžimai bei suformuluojami ir įrodomi teiginiai, kuriais remiasi šie apribojimai.

Apibrėžimas 24. Formulė G yra formulės F **poformulis**, jei $F = G, H \rightarrow G, G \rightarrow H, H \wedge G, G \wedge H, H \vee G, G \vee H, \bullet G$, kur H – bet kokia netuščia formulė, $\bullet \in \{\neg, \circ, \Box, \mathbf{B}, \mathbf{D}, \mathbf{I}\}$. Jei formulė G yra formulės F poformulis, tai bet koks formulės G poformulis taip pat yra ir formulės F poformulis.

Apibrėžimas 25. Jei $F = \Box G$, tai $H = \circ \Box G$ yra **išplėstinis F poformulis**.

Apibrėžimas 26. Formulė G yra **griežtas** formulės F **poformulis**, jei:

- G yra F poformulis
- G ilgis yra mažesnis už F ilgį [Bir08].

Apibrėžimas 27. F yra **ground formulė** sekvencijoje S , jei:

- $F = \circ \Box G$ ir $\Box G$ nėra jokios formulės iš S griežtas poformulis arba
- $F \neq \circ \Box G$ ir F nėra jokios formulės iš S poformulis.

Apibrėžimas 28. Formulės F **išplėstinė aibė** $\text{Ext}(F)$:

- jei $F = \circ \Box G$, tai $\text{Ext}(F) = \{\circ \Box G, \Box G\}$,
- jei $F = \Box G$, tai $\text{Ext}(F) = \{\circ \Box G, \Box G\}$,
- visais kitais atvejais $\text{Ext}(F) = \emptyset$. [Bir10]

Ground formulės ir išplėstinės aibės apibrėžimai svarbūs apibrėžiant sąlygas, kai ciklą susidarymas negalimas.

Tegul skaičiavime SBDI sekvencijos išvedimo paieškos medyje egzistuoja taisyklės T taikymas, kurio prielaidos – sekvencija S_1 , išvados – sekvencija S_0 . Yra žinoma, kad sekvenciniame skaičiavime BDI logikai (taigi ir SBDI) sekvencijos S_0 ir S_1 negali priklausyti jokiam ciklui, jeigu:

- S_0 yra ground formulė F , kurios nėra S_1 .
- S_1 nėra ground formulės $F' \in \text{Ext}(F)$. [Bir10]

Apibrėžimas 29. Sekvencija S yra **•-sekvencija**, kur $• \in \{\circ, \mathbf{B}\}$, jei:

1. S turi protėvį S^* , kuriam buvo taikyta modalinė taisyklė operatoriui $• ((WBEL), (W\circ))$.
2. Tarp S ir S^* nėra daugiau modalinių taisyklių su integruotu silpninimu taikymų.

Apibrėžimas 30. **•-primarinė sekvencija**, kur $• \in \{\circ, \mathbf{B}\}$, yra ir primarinė sekvencija, ir **•-sekvencija**.

Lema 1. Tegul sekvencija S yra taisyklės $(WBEL)$ taikymo išvada, o sekvencija S_1 – prielaida. Tuomet sekvencijoje S_1 visos ground formulės yra formos **BF**.

Lemos 1 įrodymas:

Sekvencijoje S_1 yra 2 tipų formulės:

1. Visos formulės **BF** iš sekvencijos S (taikant taisyklę $(WBEL)$ formulės su išoriniu operatoriumi $\mathbf{B}$ nėra nuvalomos).
2. Formulės, kurios buvo gautos nuvalius išorinį operatorių $\mathbf{B}$ nuo kokios nors formulės **BF** iš sekvencijos S .

Jokių kitų formulių negali būti, nes formulės, nesuvaržytos operatoriumi $\mathbf{B}$, taikant taisyklę $(WBEL)$ yra nuvalomos. Visos 2-o tipo formulės yra 1-o tipo formulių griežti poformuliai, todėl negali būti ground formulės. Vadinasi sekvencijoje S_1 visos ground formulės yra formos **BF**. QED.

Teiginys 3. Tegul S_p - $\circ$ -primarinė sekvencija, S_0 – artimiausias S_p protėvis, kuriam buvo taikyta taisyklė $(W \circ)$, S_1 – sekvencija, gauta pritaikius taisyklę $(WBEL)$ sekvencijai S_p . Skaičiavime SBDI negali būti ciklo $S_c \rightarrow S_c^*$, kurio pradžia S_c yra S_0 ar kuris nors S_0 protėvis, o pabaiga $S_c^* - S_1$ ar kuri nors sekvencija, kurios vienas iš protėvių yra S_1 .

Teiginio 3 įrodymas:

1. Tegul skaičiavime SBDI kokios nors sekvencijos išvedimo paieškos medyje susidarė ciklas $S_c \rightarrow S_c^*$, aprašytas teiginyje 3. Žemiau pateikiama tokią situaciją iliustruojanti schema:

$$\begin{array}{c}
\dots \\
\hline
\dfrac{S_1 : \Gamma_2, \mathbf{B}\Gamma_2 \vdash^P Q, \mathbf{B}Q, \mathbf{B}\Delta_2}{S_p : \Gamma'_2, \mathbf{B}\Gamma_2 \vdash \mathbf{B}Q, \mathbf{B}\Delta_2, \Delta'_2} \text{ (WBEL)} \\
\hline
\dots \\
\hline
\dfrac{\Gamma_1 \vdash^P \Delta_1}{S_0 : \Gamma'_1, \circ\Gamma_1 \vdash \circ\Delta_1, \Delta'_1} \text{ (W}\circ\text{)}
\end{array}$$

Pav. 14. Schema, vaizduojanti taisyklės (WBEL) taikymą $\circ$ -sekvencijai.

Pagal prielaidą, ciklo pradinė sekvencija S_c yra S_0 arba kuris nors S_0 protėvis, ciklo pabaigos sekvencija S_c^* yra S_1 arba kuri nors sekvencija S , kurios vienas iš protėvių yra S_1 .

2. Tegul sekvencijoje S_p yra bent viena ground formulė G , kurios išorinis operatorius nėra B . Pagal lemą 1, sekvencijoje S_1 visos ground formulės yra formos $\mathbf{B}F$, vadinasi sekvencijoje S_1 nebėra ground formulės G . Taip pat, pagal išplėstinės aibės apibrėžimą, jokia formulė $\mathbf{B}F$ negali priklausyti jokiai išplėstinai aibei $\text{Ext}(F)$. Vadinasi ciklo susidarymo negalimumo sąlygos yra išpildytos ir tokiu atveju žemiau S_1 ciklo būti negali.
3. Tegul S_p visos ground formulės yra pavidalo $\mathbf{B}F$. Kadangi neįmanoma prie formulės pridėti išorinio operatoriaus $\mathbf{B}$, tai visos šios formulės egzistavo visuose sekvencijos S_p protėviuose, taigi, ir sekvencijoje S_0 (galbūt kaip poformuliai).
4. Pritaikius taisyklę $(W \circ)$ sekvencijai S_0 , gautoje sekvencijoje liko tik tos formulės, kurių išorinis operatorius buvo $\circ$. Todėl kiekvienai ground formulei $G = \mathbf{B}F$ iš S_p egzistuoja formulė $\circ H$ iš S_0 , kur $G = \mathbf{B}F$ yra H poformulis.
5. Tegul $G = \mathbf{B}F$ yra kuri nors sekvencijos S_1 ground formulė, $\circ H$ - formulė iš S_0 , kurios poformulis yra G :
 - a. Visuose sekvencijos S_0 protėviuose egzistuoja formulė $\circ H$ arba formulė $\Box H$, jei $\circ H = \Box H$ (G taip pat yra H poformulis).
 - b. Sekvencijoje S_1 ir virš jos prie ground formulės G tikrai nebus pridėti išoriniai operatoriai $\circ, \Box$ nes jokia taisyklė šių operatorių negali pridėti prie formulės $G = \mathbf{B}F$.
6. Vadinasi sekvencijoje S_0 ir visuose jos protėviuose yra bent viena formulė, kurios nėra S_1 ir negali atsirasti šakoje virš S_1 . Jei $S_c \rightarrow S_c^*$ būtų ciklas (kur tarp S_c ir S_c^* yra sutinkamos sekvencijos S_0 ir S_1), tai visos formulės esančios sekvencijoje S_c būtų sutinkamos ir sekvencijoje S_c^* . Kadangi taip nėra, tai joks ciklas, prasidedantis sekvencija S_0 ar bet kuriuo jos protėviu, neužsibaigs sekvencija S_1 ar bet kokia sekvencija S , kurios vienas iš protėvių yra S_1 . QED.

Išvada 2. $\circ$ -(primarinei) sekvencijai pritaikius taisyklę (WBEL) ciklą žemiau paskutinio taisyklės $(W \circ)$ taikymo prielaidų būti negali.

Žemiau pateikiama schema, iliustruojanti išvadą 2.

$$\begin{array}{c}
 \dots \\
 \hline
 \frac{S_1 : \Gamma_2, \mathbf{B}\Gamma_2 \vdash G, \mathbf{B}G, \mathbf{B}\Delta_2}{S_p : \Gamma'_2, \mathbf{B}\Gamma_2 \vdash \mathbf{B}G, \mathbf{B}\Delta_2, \Delta'_2} (WBEL) \\
 \hline
 \dots \\
 \hline
 \frac{\Gamma_1 \vdash \Delta_1}{S_0 : \Gamma'_1, \circ\Gamma_1 \vdash \circ\Delta_1, \Delta'_1} (W\circ)
 \end{array}$$

Pav. 15. Schema, iliustruojanti apribojimą ciklams, pateiktą išvadoje 2.

Pagal išvadą 2, jei ciklo pabaiga yra sekvencija S_1 ar bet kuri vėlesnė sekvencija (bet kuri sekvencija, kurios vienas iš protėvių yra S_1), tai ciklo pradžia negali būti žemiau raudonai pažymėtos sekvencijos. Dėl to $\circ$ -(primarinei) sekvencijai pritaikius taisyklę ($WBEL$), ciklą paieška gali būti apribota - išvedus sekvenciją S_1 (ir kitas, tolimesnes sekvencijas šioje šakoje) sekvencijos S_0 ir jos protėvių tikrinti nebereikia.

Teiginys 4. B-(primarinėje) sekvencijoje bent viena ground formulė yra formos $\mathbf{B}F$.

Teiginio 4 įrodymas:

1. Jeigu turime $\mathbf{B}$ -primarinę sekvenciją, tai šakoje paskutinė taikyta modalinė taisyklė su integruotu silpninimu yra ($WBEL$).

$$\begin{array}{c}
 \frac{S_p : \Pi, \circ\Gamma_1, \mathbf{B}\Gamma_2, \mathbf{I}\Gamma_3, \mathbf{D}\Gamma_4 \vdash \Sigma, \circ\Delta_1, \mathbf{B}\Delta_2, \mathbf{I}\Delta_3, \mathbf{D}\Delta_4}{\dots} \\
 \hline
 \frac{S : \Gamma_2, \mathbf{B}\Gamma_2 \vdash^P G, \mathbf{B}G, \mathbf{B}\Delta_2}{S_0 : \Gamma'_2, \mathbf{B}\Gamma_2 \vdash \mathbf{B}G, \mathbf{B}\Delta_2, \Delta'_2} (WBEL)
 \end{array}$$

Pav. 16. $\mathbf{B}$ -primarinė sekvencija.

Pateiktoje iliustracijoje:

- S_p – nagrinėjama $\mathbf{B}$ -primarinė sekvencija.
- S_0 – artimiausias S_p protėvis, kuriam buvo taikyta taisyklė ($WBEL$).

2. Pagal lemą 1, pritaikius taisyklę (*WBEL*) gautoje sekvencijoje *S* bent viena ground formulė yra formos **BF**, nes visos ground formulės yra formos **BF** ir sekvencijoje tikrai yra bent viena ground formulė.
3. Tarp *S* ir nagrinėjamos **B**-primarinės sekvencijos *S_p* nebuvo taikomos jokios modalinės taisyklės su integruotu silpninimu.
4. Netaikant modalinių taisyklių su integruotu silpninimu formulės **BF** negalėjo būti nuvalytos ar pakeistos (prie jų negalėjo būti pridėtas joks išorinis operatorius ir nuo jų negalėjo būti nuvalytas išorinis operatorius **B**), nes:
 - a. Silpninimo taisyklės skaičiavime nėra.
 - b. Jokia kita taisyklė, išskyrus (*WBEL*), negali būti taikoma formulei **BF**.
5. Visos ground formulės **BF** iš sekvencijos *S* bus ground formulės ir sekvencijoje *S_p*, kadangi nepridedant prie jų išorinių operatorių jos netaps jokios kitos formulės poformuliais.
6. Vadinasi visos ground formulės **BF** iš sekvencijos *S* bus ground formulės ir sekvencijoje *S_p*. QED.

Pastaba: teiginys 4 galioja ne tik **B**-primarinei, bet ir bet kokiai **B**-sekvencijai. Įrodyme nagrinėjamas **B**-primarinės sekvencijos atvejis, nes šiame darbe pateikiami sekvenciniai skaičiavimai, kuriuose modalinės taisyklės su integruotu silpninimu taikomos tik primarinėms sekvencijoms. Taip pat teisingas ir teiginys, kad **B**-primarinėje sekvencijoje visos ground formulės yra formos **BF**, tačiau šiame darbe pateikiamiems apribojimams užtenka fakto, kad yra bent viena tokia formulė.

Išvada 3. Taikant taisyklę (*W* ○) **B**-primarinei sekvencijai nuvaloma bent viena ground formulė **BF**. Taip pat bet kokiai $F \text{ Ext}(\mathbf{BF}) = \emptyset$, todėl taisyklės (*W* ○) taikymo prielaidose tikrai nėra tokios formulės *G*, kuri priklausytų kokios nors formulės **BF** išplėstinei aibei. Vadinasi žemiau taisyklės (*W* ○) taikymo **B**-primarinei sekvencijai prielaidų ciklą būti negali.

Žemiau pateikiama išvadą 3 iliustruojanti schema.

$$\begin{array}{c}
 \dots \\
 \hline
 S^* : \Gamma_1 \vdash^P \Delta_1 \\
 \hline
 S_p : \Pi, \circ\Gamma_1, \mathbf{B}\Gamma_2, \mathbf{I}\Gamma_3, \mathbf{D}\Gamma_4 \vdash \Sigma, \circ\Delta_1, \mathbf{B}\Delta_2, \mathbf{I}\Delta_3, \mathbf{D}\Delta_4 \quad (W \circ) \\
 \hline
 \dots \\
 \hline
 S : \Gamma_2, \mathbf{B}\Gamma_2 \vdash^P G, \mathbf{B}G, \mathbf{B}\Delta_2 \\
 \hline
 S_0 : \Gamma'_2, \mathbf{B}\Gamma_2 \vdash \mathbf{B}G, \mathbf{B}\Delta_2, \Delta'_2 \quad (WBEL)
 \end{array}$$

Pav. 17. Schema, iliustruojanti apribojimą ciklams, pateiktą išvadoje 3.

Schemoje raudonai pažymėta sekvencija S^* . Žemiau šios sekvencijos ciklą būti negali, nes taikant taisyklę $(W \circ)$ buvo nuvalyta bent viena sekvencijos S_p ground formulė **BF**, kuri niekaip nebeatsiras sekvencijoje S^* ar virš jos.

Pabrėžtina, kad šiame skyriuje pateikti apribojimai laiko ir įsitikinimų ciklams yra skirtingi. Kai **B**-primarinei sekvencijai pritaikoma taisyklė $(W \circ)$, ciklo žemiau šios taisyklės taikymo prielaidų būti negali, nes taikant taisyklę $(W \circ)$ nuo taisyklės taikymo išvadų nuvaloma bent viena formulė, kurios nebeįmanoma gauti aukščiau taisyklės $(W \circ)$ taikymo prielaidų. Tuo tarpu kai $\circ$ -primarinei sekvencijai pritaikoma taisyklė $(WBEL)$, apribojimas ciklams nėra toks griežtas. Ciklai vis dar gali susidaryti, tik jų pradžia negali būti žemiau artimiausio (keliaujant iš viršaus į apačią) taisyklės $(W \circ)$ taikymo prielaidų. Taip yra, nes šiuo atveju abiejų taisyklių taikymai (be kitų modalinių taisyklių su integruotu silpninimu taikymų tarp jų) yra reikalingi, kad iš taisyklės $(W \circ)$ taikymo išvadų būtų nuvalytos formulės, kurios nebegalės atsirasti aukščiau taisyklės $(WBEL)$ taikymo išvadų.

Iš šiame skyriuje įrodytų teiginių 2, 3, 4 seka, kad negali būti ciklo, kurio viduje yra ir taisyklės $(W \circ)$, ir taisyklės $(WBEL)$ taikymai.

Sekvencijų išvedimo pavyzdžiai, iliustruojantys šiame poskyryje pateiktus apribojimus pateikiami žemiau, bet prieš tai apibrėžiamas sekvencinis skaičiavimas, kuriame šie apribojimai panaudojami ciklų paieškos efektyvumui padidinti.

6.4. Sekvencinis skaičiavimas SBDI1 su procedūra

Šiame poskyryje apibrėžiamas sekvencinis skaičiavimas SBDI1 su procedūra P1. Šiame skaičiavime ir procedūroje panaudojami anksčiau šiame skyriuje apibrėžti apribojimai ciklams, todėl gaunamas skaičiavimas, kuriame ciklų aptikimas yra efektyvesnis nei klasikiniame sekvenciniame skaičiavime BDI logikai.

Apibrėžimas 31. Sekvencinis skaičiavimas SBDI1 – tai sekvencinis skaičiavimas, kuriame yra aksioma $\Gamma, A \vdash \Delta, A$, ciklinė aksioma, loginės taisyklės:

$$\begin{array}{ll}
 (\rightarrow \vdash) \quad \frac{\Gamma \vdash [\alpha] \Delta, F \quad \Gamma, G \vdash [\alpha] \Delta}{\Gamma, F \rightarrow G \vdash [P, \alpha, \bullet] \Delta} & (\rightarrow \rightarrow) \quad \frac{\Gamma, F \vdash [\alpha] \Delta, G}{\Gamma \vdash [P, \alpha, \bullet] \Delta, F \rightarrow G} \\
 (\& \vdash) \quad \frac{\Gamma, F, G \vdash [\alpha] \Delta}{\Gamma, F \& G \vdash [P, \alpha, \bullet] \Delta} & (\vdash \&) \quad \frac{\Gamma \vdash [\alpha] \Delta, F \quad \Gamma \vdash [\alpha] \Delta, G}{\Gamma \vdash [P, \alpha, \bullet] \Delta, F \& G} \\
 (\vee \vdash) \quad \frac{\Gamma, F \vdash [\alpha] \Delta \quad \Gamma, G \vdash [\alpha] \Delta}{\Gamma, F \vee G \vdash [P, \alpha, \bullet] \Delta} & (\vdash \vee) \quad \frac{\Gamma \vdash [\alpha] \Delta, F, G}{\Gamma \vdash [P, \alpha, \bullet] \Delta, F \vee G}
 \end{array}$$

$$(\neg \vdash) \quad \frac{\Gamma \vdash^{[\alpha]} \Delta, F}{\Gamma, \neg F \vdash^{[P, \alpha, \bullet]} \Delta}$$

$$(\vdash \neg) \quad \frac{\Gamma, F \vdash^{[\alpha]} \Delta}{\Gamma \vdash^{[P, \alpha, \bullet]} \Delta, \neg G}$$

Ir modalinės taisyklės:

$$(WBEL) \quad \frac{\Gamma, \mathbf{B}\Gamma \vdash^{[p, B]} G, \mathbf{B}G, \mathbf{B}\Delta}{\Gamma', \mathbf{B}\Gamma \vdash^{[P, \alpha, \bullet]} \Delta', \mathbf{B}G, \mathbf{B}\Delta}$$

$$(WDES^{\wedge}) \quad \frac{\Gamma \vdash^{[p, \wedge]} G}{\Gamma', \mathbf{D}\Gamma \vdash^{[P, \alpha, \bullet]} \Delta', \mathbf{D}G, \mathbf{D}\Delta}$$

$$(\Box \vdash) \quad \frac{F, \Box F, \Gamma \vdash^{[\alpha]} \Delta}{\Box F, \Gamma \vdash^{[P, \alpha, \bullet]} \Delta}$$

$$(W \circ B) \quad \frac{\Gamma \vdash^{[p, \wedge]} \Delta}{\Gamma', \circ \Gamma \vdash^{[P, B, \bullet]} \Delta, \Delta'}$$

$$(WINT^{\wedge}) \quad \frac{\Gamma \vdash^{[p, \wedge]} G}{\Gamma', \mathbf{I}\Gamma \vdash^{[P, \alpha, \bullet]} \Delta', \mathbf{I}G, \mathbf{I}\Delta}$$

$$(\vdash \Box) \quad \frac{\Gamma \vdash^{[\alpha, -]} \Delta, F \quad \Gamma \vdash^{[\alpha, +]} \Delta, \Box F}{\Gamma \vdash^{[P, \alpha, \bullet]} \Delta, \Box F}$$

$$(W \circ) \quad \frac{\Gamma \vdash^{[p, \circ]} \Delta}{\Gamma', \circ \Gamma \vdash^{[P, \alpha', \bullet]} \Delta, \Delta'}$$

Čia $\alpha \in \{\emptyset, \circ, B, \wedge\}$, $\alpha' \in \{\emptyset, \circ, \wedge\}$, $\bullet \in \{\emptyset, +, -\}$, $P \in \{p, \emptyset\}$.

Naujo skaičiavimo SBDI1 pagrindas yra anksčiau apibrėžtas skaičiavimas SBDI. Naujame skaičiavime atlikti tokie pakeitimai:

1. Skaičiavime SBDI1 naudojamos žymėtos sekvencijos. Kiekviena sekvencija turi savo žymę $[P, \alpha, \bullet]$.
 - a. Sekvencijos žymėje $P \in \{p, \emptyset\}$ nurodo ar sekvencija gauta pritaikius modalinę taisyklę su integruotu silpninimu primarinei sekvencijai ir ciklą paieškos metu yra lygintina su naujai gauta sekvencija, kur p žymi, kad sekvencija yra lygintina su naujai gauta sekvencija.
 - b. Sekvencijos žymėje $\alpha \in \{\emptyset, \circ, B, \wedge\}$ nurodo ar sekvencija yra $\circ$ -sekvencija, **B**-sekvencija ar ciklą paieškos pabaigos sekvencija, kur $\circ$ žymi, kad sekvencija yra $\circ$ -sekvencija, B – kad sekvencija yra **B**-sekvencija, $\wedge$ - kad sekvencija yra ciklą paieškos pabaigos sekvencija (žemiau šios sekvencijos ciklo būti negali, nes ši sekvencija negali būti jokio ciklo viduje).
 - c. Sekvencijos žymėje $\bullet \in \{\emptyset, +, -\}$ nurodo ar sekvencija yra kuri nors taisyklės $(\vdash \Box)$ prielaida, kur $+$ žymi, kad sekvencija yra dešinioji šios taisyklės prielaida, $-$ žymi, kad sekvencija yra kairioji šios taisyklės prielaida. Žymėjimas $\bullet$ įvedamas ciklinės aksiomos identifikavimui: vykdant ciklą paiešką pakanka sekti

ar pakeliui buvo sutiktos + ir – pažymėtos sekvencijos. Aptiktas ciklas yra ciklinė aksioma tik jeigu ciklo paieškos metu buvo aptikta bent viena sekvencija, pažymėta +, ir neaptikta nei viena sekvencija, pažymėta –.

2. Taisyklė $(W \circ)$ šiame skaičiavime pakeista dviem taisyklėmis: $(W \circ)$ ir $(W \circ B)$. Taisyklė $(W \circ B)$ gali būti taikoma tik **B**-sekvencijai ir pažymi gautą sekvenciją ^, nes, pagal išvadą 3, pritaikius **B**-primarinei sekvencijai taisyklę $(W \circ)$ žemiau ciklų būti negali.

Taip pat skaičiavimui SBDI1 anksčiau apibrėžtos procedūros P pagrindu žemiau apibrėžiama nauja procedūra P1.

Apibrėžimas 32. Procedūra P1 sekvenciniam skaičiavimui SBDI1 – tai procedūra, gauta iš procedūros P sekvenciniam skaičiavimui SBDI, pakeičiant 4-ą procedūros žingsnį tokiu:

4. Jei eilė Q netuščia, iš eilės imamas taisyklės T taikymas, jį pritaikius sekvencijai S_p gaunama sekvencija $S_p^`$ . Patikrinama, ar išvedus  $S_p^`$ susidarė ciklas - $S_p^`$, keliaujant iš viršaus žemyn, lyginama su savo protėviais, pažymėtais $\vdash^P$ žymėjimu, iki
 - pirmos $\circ$ -sekvencijos, jei tikrinama **B**-sekvencija (kadangi tikrinamos tik iš primarinių gautos sekvencijos, pirmoji $\circ$ -sekvencija ir bus artimiausio iš viršaus žemyn taisyklės $(W \circ)$ taikymo prielaidos) arba
 - Pirmos sekvencijos, pažymėtos „^“ ženklu (jei $S_p^`$ pažymėta ^ ženklu, ciklų paieška iš viso nevykdoma):
 - a. Jei išvedus sekvenciją $S_p^`$ aptikta ciklinė aksioma (cikle aptinkama sekvencija, pažymėta + ir neaptinkama sekvencijų, pažymėtų -), grąžinamas 1.
 - b. Jei aptinkamas ciklas, kuris nėra ciklinė aksioma, bandomas kitas taikymas iš Q - grįžtama į 4-ą žingsnį.
 - c. Jei ciklas neaptiktas, sekvencijai $S_p^`$  kviečiama procedūra P (keliaujama į 1-ą žingsnį). Jei  $S_p^`$ – išvedama, grąžinamas 1.

Pastaba: procedūroje P1 taisyklės taikomos tokia pačia tvarka kaip procedūroje P (dėl taisyklės $(W \circ B)$ naujos taikymo tvarkos apibrėžti nereikia, nes kai galima taikyti taisyklę $(W \circ B)$, negalima taikyti taisyklės $(W \circ)$ ir atvirkščiai).

Iš įrodytų teiginių 2, 3, 4 seka, kad naujasis skaičiavimas SBDI1 su procedūra P1 yra ekvivalentus anksčiau apibrėžtam skaičiavimui SBDI su procedūra P. Skaičiavime SBDI1 visos taisyklės turi tikslius atitikmenis taisyklėms iš SBDI (nežiūrint į naujų sekvencijų papildomus žymėjimus, kurie aktualūs tik ciklų paieškos metu). Vienintelė išimtis – į 2 dalis padalinta taisyklė $(W \circ)$, tačiau skaičiavime SBDI1 esančios 2 taisyklės $(W \circ B)$, $(W \circ)$ kartu yra ekvivalenčios taisyklei $(W \circ)$ iš SBDI. $(W \circ B)$ taikoma tik **B**-sekvencijoms, $(W \circ)$ - visoms likusioms, o naujai gauta sekvencija yra identiška sekvencijai, kuri būtų gauta taikant taisyklę $(W \circ)$ iš skaičiavimo SBDI (nežiūrint sekvencijų į žymėjimus, kurie aktualūs tik ciklų paieškos metu).

Naujai apibrėžta procedūra P1 nuo procedūros P taip pat skiriasi tik ciklų paieškos aspektu, nes taisyklių taikymo tvarka yra nepakitusi. Taigi naujasis skaičiavimas SBDI1 su procedūra P1 yra ekvivalentus klasikiniam sekvenciniam skaičiavimui BDI logikai ir baigtinis, kaip ir skaičiavimas SBDI.

Atmetus sekvencijų žymėjimo skirtumus abiejuose skaičiavimuose gaunami identiški sekvencijos išvedimo paieškos medžiai, skiriasi tik vykdoma ciklų paieška, kuri skaičiavime SBDI1 visada yra tokia pat efektyvi arba efektyvesnė nei skaičiavime SBDI.

6.5. Sekvencijų išvedimo ir ciklų paieškos pavyzdžiai skaičiavime SBDI1

Šiame poskyryje pateikiami sekvencijų išvedimo pavyzdžiai skaičiavime SBDI1.

$$\begin{array}{c}
\frac{(ax)}{q, \mathbf{B}q \vdash^{[\mathbf{B},]} q, r, \mathbf{B}(q \vee r)} \quad \frac{}{q \vee r, \mathbf{B}(q \vee r)} (\vee \vdash) \\
\frac{}{S_3 : q, \mathbf{B}q \vdash^{[p, \mathbf{B},]} q \vee r, \mathbf{B}(q \vee r)} (WBEL) \\
\frac{}{p, \mathbf{B}q \vdash^{[p, \circ,]} \mathbf{B}(q \vee r)} (\wedge \vdash) \\
\frac{(ax)}{r, \circ \Box r \vdash^{[\circ,]} r, \circ \mathbf{B}(q \vee r)} \quad \frac{}{r, \circ \mathbf{B}(q \vee r)} (\Box \vdash) \\
\frac{}{\Box r \vdash^{[\circ,]} r, \circ \mathbf{B}(q \vee r)} (\vdash \vee) \\
\frac{}{\Box r \vdash^{[\circ,]} r \vee \circ \mathbf{B}(q \vee r)} (\vee \vdash) \\
\frac{}{S_1 : \Box r \vee \circ(p \wedge \mathbf{B}q) \vdash^{[p, \circ,]} r \vee \circ \mathbf{B}(q \vee r)} (W\circ) \\
\frac{}{\circ(\Box r \vee \circ(p \wedge \mathbf{B}q)) \vdash^{[,,]} \circ(r \vee \circ \mathbf{B}(q \vee r)), q, r} (\vdash \vee) \\
\frac{}{\circ(\Box r \vee \circ(p \wedge \mathbf{B}q)) \vdash^{[,,]} \circ(r \vee \circ \mathbf{B}(q \vee r)), q \vee r} (\vdash \vee)
\end{array}$$

Pav. 18. Sekvencijos išvedimo medžio, kuriame $\circ$ -sekvencijai taikoma taisyklė ($WBEL$), pavyzdys.

Pavyzdyje vykdant ciklą paiešką sekvencijai S_3 ar virš jos esančioms sekvencijoms, žemiau raudonai pažymėtos sekvencijos S_2 ciklą tikrinti nereikia. Ciklą žemiau būti negali, nes S_3 gauta pritaikius taisyklę ($WBEL$) $\circ$ -primarinei sekvencijai, o S_2 yra artimiausias S_3 protėvis, gautas pritaikius taisyklę ($W\circ$).

Pavyzdyje vykdant ciklą paiešką be apribojimų išvedus S_3 ši sekvencija būtų lyginama su S_2 ir S_1 . Tuo tarpų taikant apribojimą ciklą paieška baigtusi patikrinus sekvenciją S_2 ir palyginimo su S_1 būtų išvengta.

Žemiau pateikiamas dar vienas sekvencijos išvedimo medžio pavyzdys.

$$\begin{array}{c}
\frac{(ax)}{q \vdash^{[., \circ, .]} q, r} \\
\frac{\frac{\frac{S_3 : q \vdash^{[p, \circ, .]} q \vee r}{\circ q \vdash^{[., \circ, .]} \circ(q \vee r), s} (\vdash \vee)}{S_2 : \circ q \vdash^{[p, \wedge, .]} \circ(q \vee r) \vee s} (W \circ) \\
\frac{\frac{S : p, \circ \circ q, \mathbf{B}(p \wedge \circ \circ q) \vdash^{[., \mathbf{B}, .]} \circ(\circ(q \vee r) \vee s), \mathbf{B}(\circ(\circ(q \vee r) \vee s))}{S_1 : p \wedge \circ \circ q, \mathbf{B}(p \wedge \circ \circ q) \vdash^{[p, \mathbf{B}, .]} \circ(\circ(q \vee r) \vee s), \mathbf{B}(\circ(\circ(q \vee r) \vee s))} (\wedge \vdash)}{\mathbf{B}(p \wedge \circ \circ q) \vdash^{[., .]} \mathbf{B}(\circ(\circ(q \vee r) \vee s))} (WBEL)
\end{array}$$

Pav. 19. Sekvencijos išvedimo medžio, kuriame **B**-sekvencijai taikoma taisyklė $(W \circ B)$, pavyzdys.

Pateiktame pavyzdyje raudonai pažymėta sekvencija S_2 , žemiau kurios ciklą nebegali būti (tai nurodo ir $\wedge$ žymėjimas). Žemiau pažymėtos sekvencijos ciklą būti nebegali, nes ši sekvencija gauta pritaikius $(W \circ B)$ taisyklę **B**-sekvencijai, o toks taikymas, kaip buvo įrodyta, nuvalo ground formules **BF** (pavyzdyje abi formulės su išoriniais operatoriais **B** yra ground formulės, nes nėra tokios formulės, kurios griežtas poaibis būtų kuri nors iš šių formulių).

Vykdam ciklą paiešką be apribojimų išvedus sekvenciją S_3 ji turėtų būti lyginama su sekvencijomis S_2 ir S_1 . Pritaikius apribojimą būtų tikrinama tik sekvencija S_2 ir, kadangi ši sekvencija pažymėta $\wedge$ žymėjimu, ciklą paieška žemiau nebūtų vykdoma.

Pastebėtina, kad sekvencijai S galima taikyti ir taisyklę $(WBEL)$. Tuomet būtų gaunamas sekvencijos išvedimo paieškos medis, vaizduojamas žemiau.

$$\begin{array}{c}
(C) \\
\frac{\frac{S^* : p \wedge \circ \circ q, \mathbf{B}(p \wedge \circ \circ q) \vdash^{[p, \mathbf{B}, .]} \circ(\circ(q \vee r) \vee s), \mathbf{B}(\circ(\circ(q \vee r) \vee s))}{S : p, \circ \circ q, \mathbf{B}(p \wedge \circ \circ q) \vdash^{[., \mathbf{B}, .]} \circ(\circ(q \vee r) \vee s), \mathbf{B}(\circ(\circ(q \vee r) \vee s))} (WBEL)}{\frac{S_1 : p \wedge \circ \circ q, \mathbf{B}(p \wedge \circ \circ q) \vdash^{[p, \mathbf{B}, .]} \circ(\circ(q \vee r) \vee s), \mathbf{B}(\circ(\circ(q \vee r) \vee s))}{\mathbf{B}(p \wedge \circ \circ q) \vdash^{[., .]} \mathbf{B}(\circ(\circ(q \vee r) \vee s))} (\wedge \vdash)} (WBEL)
\end{array}$$

Pav. 20. Alternatyvus sekvencijos išvedimo paieškos medis.

Šiame pavyzdyje išvedus sekvenciją S^* ji būtų lyginama su sekvencija S_1 . Kadangi sekvencijos lygios, šaka užsibaigia ciklu (nėra išvedama). Laikantis procedūroje P nurodytos taisyklių taikymo tvarkos šis medis nebūtų sukonstruotas, nes pirmiausia būtų taikoma taisyklė $(W \circ B)$ ir sekvencija būtų sėkmingai išvesta (būtų sukonstruotas anksčiau pavaizduotas

sekvencijos išvedimo medis). Tačiau šis pavyzdys parodo, kad automatinei priemonei, kuri įgyvendina skaičiavimą su procedūra, gali tekti konstruoti kelis sekvencijos išvedimo paieškos medžius net kai sekvencija išvedama (tam pakanka vienintelio sekvencijos išvedimo medžio, kuriame visos šakos užsidaro aksiomomis). Šis pavyzdys taip pat iliustruoja skaičiavimo sudėtingumo ir sekvencijų palyginimų skaičiaus įvertinimo problematiškumą.

Bendru atveju sunku formaliai įvertinti kiek palyginimų teks atlikti sekvencijos išvedimo paieškos medžio konstravimo metu skaičiavime SBDI1 (ar SBDI). Tačiau darbe suformuluoti ir įrodyti apribojimai lemia tai, kad išvedimo paieška skaičiavime SBDI1 tikrai negali būti sudėtingesnė nei skaičiavime SBDI. Pateikti pavyzdžiai parodo, kad kai kuriais atvejais ciklą paieška (taigi ir sekvencijos išvedimo medžio konstravimas) tikrai yra efektyvesnė. Didesnis efektyvumo padidėjimas būtų pasiekiamas sudėtingesnėms sekvencijoms, kurių išvedimo medžiai yra dideli ir netinkami pateikti darbe.

Realiuose sekvencijų pavyzdžiuose dažnai yra labai daug formulų, suvaržytų įvairiais modaliniais operatoriais. Šių sekvencijų išvedimo paieškos medžiuose tarp šaknies ir lapų daug kartų taikomos modalinės taisyklės. Naudojant skaičiavimą SBDI kiekvieną naują sekvenciją, pažymėtą $\vdash^p$ žymėjimu, tektų lyginti su visais taip pažymėtais protėviais. Skaičiavime SBDI1 beveik visais atvejais apsiribojama palyginimu su artimiausiu tokiu protėviu. Išimtis yra tik ciklai, kuriuose yra netaikomos (*WBEL*), (*WDES*), (*WINT*) taisyklės, susidarantys vien dėl laiko modalinių taisyklių ($(W \circ)$, $(\Box \vdash)$, $(\vdash \Box)$ taikymų.

Skaičiavimas SBDI1 leidžia apriboti ciklą paiešką, kai sekvencijoje yra formulų su įvairiais modaliniais operatoriais. Praktikoje ilgesni ciklai, kuriuose kelis kartus taikomos tik $(W \circ)$, $(\Box \vdash)$, $(\vdash \Box)$ taisyklės, susidaro gerokai rečiau nei trumpi ciklai, kuriems dėl naudojamų apribojimų skaičiavime SBDI1 dažniausiai užtektų palyginti einamąją sekvenciją tik su vienu iš protėvių (skaičiavime SBDI be apribojimų dažnai tektų lyginti su daugeliu protėvių). Taigi skirtingų operatorių tarpusavio sąveika leidžia išvengti dalies sekvencijų palyginimų ir padidinti skaičiavimo efektyvumą.

7. Rezultatai ir išvados

Atliekant šį darbą pasiekti tokie rezultatai:

1. Apibrėžtas skaičiavimas SBDI be silpninimo taisyklės ir su silpninimu integruotu į modalines taisykles (*WBEL*), (*WDES*), (*WINT*), ($W \circ$).
2. Apibrėžta baigtinė procedūra P, nustatanti ar sekvencija S išvedama skaičiavime SBDI.
3. Įrodytas skaičiavimų SBDI* (įprasto sekvencinio skaičiavimo BDI logikai su tiesine laiko logika) ir SBDI ekvivalentumas.
4. Suformuluoti ir įrodyti apribojimai ciklų susidarymui skaičiavime SBDI ir pateikta pavyzdžių, rodančių, kad apribojimai padidina ciklų paieškos efektyvumą, taigi, ir išvedimo paieškos efektyvumą kai kurioms sekvencijoms.
5. Remiantis įrodytais apribojimais ir skaičiavimu SBDI su procedūra P apibrėžtas sekvencinis skaičiavimas SBDI1 su baigtine procedūra P1 sekvencijos išvedamumui šiame skaičiavime patikrinti. Skaičiavime SBDI1 ciklų paieška yra apribota ir dėl to efektyvesnė.

Atlikus šį darbą darytinos tokios išvados:

1. Darbe įrodyti apribojimai ciklų susidarymui sekvenciniame skaičiavime SBDI leidžia apriboti ciklų paiešką.
2. Skaičiavime SBDI1, lyginant su skaičiavimu SBDI, ciklų paieška yra efektyvesnė, kai pradinėje sekvencijoje yra formulų, suvaržytų skirtingais modaliniais operatoriais.
3. Multimodalinėse logikose skirtingų modalumų tarpusavio sąveika gali lemti ciklų susidarymo negalimumą ar apribojimus, taigi ir efektyvesnę ciklų paiešką. Tokių apribojimų gauti nebūtų galima nagrinėjant monomodalines logikas atskirai.

Bet kokios sekvencijos išvedimo medžiai skaičiavimuose SBDI1 ir SBDI yra identiški, todėl sekvencijos išvedimas skaičiavime SBDI1 gali būti arba toks pat efektyvus, arba efektyvesnis dėl apribotos ciklų paieškos. Šiame darbe apibrėžti apribojimai pademonstruoja, kad sekvenciniuose skaičiavimuose multimodalinėms logikoms galima identifikuoti situacijas, kai ciklai negalimi ar

riboti, nagrinėjant skirtingų modalumų (taisyklių skirtingiems modalumams) sąveikas. Darbe apibrėžtus apribojimus ciklą paieškai galima taikyti kartu su kitokiais apribojimais sekvenciniams skaičiavimams BDI logikai ar bet kuriai iš BDI logikoje naudojamų monomodalinių logikų.

Literatūra

- [Alo12] R. Alonderis. A labeled sequent calculus for propositional linear time logic. 2012, Liet. matem. rink. 53, 2012, 1-6.
- [AP13] R. Alonderis, R. Pliuškevičius. Sequent systems for PLTL, 2013, Liet. matem. rink. 54, 2013, 1-5. <https://www.journals.vu.lt/LMR/article/view/14872/13891>
- [BH06] R. Bordini, J. Hübner. BDI agent programming in AgentSpeak using Jason, 2006. Lecture Notes in Computer Science, 143-164. https://doi.org/10.1007/11750734_9.
- [Bir06a] A. Birštunas. Efficient loop-check for KD45 logic. 2006, *Lithuanian Mathematical Journal*, Vol. 46, No. 1, 2006.
- [Bir06b] A. Birštunas. Sequent calculus usage for BDI agent implementation. 2006, *Liet. matem. rink*, 46, spec. nr., 2006, 1–6.
- [Bir08] A. Birštunas. Restrictions for loop-check in sequent calculus for temporal logic. 2008, Liet. mat. rink. LMD darbai, 48/49, 269–274.
- [Bir10] A. Birštunas. Sequent calculi with an efficient loop-check for BDI logics. 2010, Vilnius.
- [DHH15] H. Ditmarsch, J. Halpern, W. Hoek, B. Kooi. An Introduction to Logics of Knowledge and Belief, 2015. 51 psl.
- [EH86] E. Emerson, J. Halpern. "Sometimes" and "not never" revisited: on branching versus linear time temporal logic, 1986. Journal of the ACM, 33(1), 151–178. <https://doi.org/10.1145/4904.4999>
- [GPP+99] M. Georgeff, B. Pell, M. Pollack, M. Tambe, M. Wooldridge. The Belief-Desire-Intention Model of Agency, 1999.
- [How98] J. Howe. Proof search issues in some non-classical logics, 1998. Prieiga internete: https://kar.kent.ac.uk/21574/2/Proof_Search_Issues_in_Some_Non-Classical_Logics.pdf
- [Nor04] S. Norgėla. Matematinė logika. TEV, Vilnius, 2004. 191 psl.
- [NS04] E. Norling, L. Sonenberg. Creating Interactive Characters with BDI Agents, 2004.
- [NZ00] U. Nodelman, E. Zalta. Modal Logic, 2000 (2023). *Stanford Encyclopedia of Philosophy*. Prieiga internete: <https://plato.stanford.edu/entries/logic-modal/#ModLog>.
- [Woo00] M. Wooldridge. Reasoning about rational agents. Intelligent robots and autonomous agents, 2000.