

Damian Heiman

Modaalilogiikka ja lokaalit algoritmit

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Matematiikka
Toukokuu 2020

TIIVISTELMÄ

Damian Heiman: Modaalilogiikka ja lokaalit algoritmit
Pro gradu -tutkielma
Tampereen yliopisto
Matematiikan ja tilastotieteen tutkinto-ohjelma
Toukokuu 2020

Tässä tutkielmassa tutkitaan modaalilogiikan yhteyksiä lokaaleihin algoritmeihin. Modaalilogiikka on lauselogiikan laajennus, jossa kaavojen totuutta tutkitaan mahdollisia maailmoja mallintavissa Kripke-malleissa. Lokaalit algoritmit ovat graafeissa ajettavia algoritmeja, jotka päättyvät vakioajassa. Algoritmeilla ratkaistaan graafiongelmia, joille esitetään luokittelu sen mukaan, voidaanko ne ratkaista vakioajassa ja miten paljon informaatiota algoritmit tarvitsevat niiden ratkaisemiseksi. Vakioaikaiset graafiongelmaluokat karakterisoidaan modaalilogiikoiden avulla samaistamalla syötegraafit sopiviin Kripke-malleihin. Tutkielma keskittyy heikkoihin hajautetun laskennan malleihin, jotka hyödyntävät graafeihin lisättäviä porttinumerointeja, ja laajentaa aiempia niitä koskevia tuloksia tarkastelemalla, miten tulokset yleistyvät yksinkertaisista graafeista suunnattuihin graafeihin. Tuloksien todistamiseen käytetään bisimulaatiota, joka liittää toisiinsa Kripke-mallit, joita modaalilogiikka ei erota toisistaan. Erityisesti osoitetaan, että graafiongelmaluokkien välinen hierarkia on suunnattujen graafien tapauksessa heikompi kuin yksinkertaisten graafien tapauksessa. Osa luokkien välisistä inklusioista ei päde, kun rajoitus yksinkertaisiin graafeihin poistetaan.

Avainsanat: modaalilogiikka, lokaali algoritmi, hajautettu laskenta, graafiongelma, porttinumerointi, bisimulaatio
Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

Sisältö

1 Johdanto	4
1.1 Tutkielman rakenne	5
2 Modaalilogiikat, Kripke-mallit ja bisimulaatio	6
2.1 Modaalilogiikat ja Kripke-mallit	6
2.2 Bisimulaatiot	8
3 Graafit, algoritmit ja graafiongelmat	12
3.1 Graafit	12
3.2 Algoritmit	14
3.3 Algoritmien luokat	15
3.4 Graafiongelmat	16
3.5 Graafiongelmien luokat	19
4 Vakioaikaisten luokkien karakterisointi modaalilogiikoilla	21
5 Graafiongelmien luokkien väliset identiteetit ja eroavaisuudet	33
5.1 Luokkien väliset identiteetit	33
5.2 Luokkien väliset eroavaisuudet	36
Lähteet	41

1 Johdanto

Hajautettu laskenta on tietoverkossa tapahtuvaa laskentaa usean laskentayksikön välillä. Teoreettisessa tietojenkäsittelytieteessä tietoverkko käsitetään usein graafina, joka koostuu solmuista ja särmistä. Laskentayksiköt tulkitaan solmuina ja niiden välinen kommunikaatio tapahtuu särmien välityksellä. Hajautettu algoritmi on tilakone, josta on kopio jokaisessa tietoverkon solmussa. Se määrää solmun tilan laskennan alussa ja sen, mihin tilaan missäkin tapauksessa siirrytään. Laskenta tapahtuu, kun solmut siirtyvät tilasta toiseen ja lähettävät toisilleen viestejä tapahtuvista muutoksista. Lokaalit algoritmit ovat hajautettuja algoritmeja, jotka suorittavat laskennan vakioajassa, riippumattomana tietoverkon koosta.

Tässä tutkielmassa tarkastelemme erityisesti laskentaa, joka on synkronista ja determinististä. Tämä tarkoittaa, että jokainen solmu suorittaa saman laskennan vaiheen samanaikaisesti ja laskennan tulos on valitussa syötegraafissa aina sama. Tarkastelumme keskittyy laskennan malleihin, jotka hyödyntävät porttinumeroita, jotka liitetään jokaisen särmän kumpaankin päähän. Porttinumeroiden avulla solmut voivat pitää kirjaa siitä, mitä viestejä on lähetetty tai vastaanotettu mitään särkeä pitkin.

Modaalilogiikka on matemaattisen logiikan osa-alue, joka laajentaa lauselogiikkaa modaalioperaattorilla \diamond . Se laajentaa kaavojen tulkintaa mahdollisten maailmojen käsitteellä: jos ϕ on lauselogiikan kaava, niin kaavan $\diamond\phi$ tulkinta on 'on olemassa mahdollinen maailma, jossa ϕ on tosi.' Tutkinnan kohteena ovat Kripke-mallit ja -kehykset. Kehykset koostuvat maailmoista (tässä tutkielmassa puhumme pisteistä) ja niiden välisistä relaatioista. Mallit ovat kehyksiä, joihin on liitetty valuaatiofunktio, joka määrää jokaiselle atomikaavalle joukon maailmoja, joissa se on tosi. Kaavojen totuutta tutkitaan mallin maailmoissa ja relaatiot kuvaavat, mitkä muut maailmat ovat mahdollisia tarkasteltavan maailman näkökulmasta.

Tutkielmassamme tarkastelemme perusmodaalilogiikkaa ja kolmea sen laajennusta: multimodaalilogiikkaa, porrastettua modaalilogiikkaa ja porrastettua multimodaalilogiikkaa. Multimodaalilogiikassa Kripke-malleissa on monta eri relaatiota ja vastaavasti monta eri modaalioperaattoria. Porrastetussa modaalilogiikassa modaalioperaattoreita on yksi, mutta siihen yhdistetään luku, joka asettaa alarajan halutunlaisille mahdollisille maailmoille. Porrastettu multimodaalilogiikka on multimodaalilogiikan ja porrastetun modaalilogiikan yhdistelmä.

Hella et al. [6] esittivät uuden keinon lähestyä hajautettuja algoritmeja modaalilogiikan näkökulmasta. Aiemmin Kripke-mallien maailmat samaistettiin algoritmin tiloiksi ja relaatiot mahdollisiksi siirtymiksi tilasta toiseen. Tätä lähestymistapaa käsitelivät esimerkiksi van Benthem, Eijck ja Stebletsova [1], modaalilogiikan, aikalogiikan ja propositionaalisen dynaamisen logiikan näkökulmista. Bisimulaation käsitettä he hyödynsivät eri tilakoneiden samankaltaisuuden osoittamiseksi: kaksi keskenään bisimilaarista tilakonetta voivat tehdä toisiaan vastaavia siirtymiä tilasta toiseen. Uudessa lähestymistavassa Kripke-mallit samaistetaan tietoverkkoon ja maailmat laskentayksiköihin, joiden välinen viestintä tapahtuu relaatioita pitkin. Tämä lähestymistapa mahdollisti erilaisten graafiongelmaluokkien karakterisoimisen eri modaalilogiikan laajennusten avulla sekä bisimulaation käytön graafiongelmiin ratkeavuuden tutkimisessa. Artikkelin tärkein tulos oli täysi esitys graafiongelmaluokkien välisistä osajoukkorelaatioista. Osoitettiin nimittäin, että ne muodostavat keskenään lineaarijärjestyksen.

Tässä tutkielmassa laajennamme näitä tuloksia yleistämällä tarkastelun algoritmeihin, joissa laskentayksiköt eivät välttämättä kykene välittämään tietoa kumpaankin suuntaan toistensa välillä. Tällaiset tietoverkot samaistuvat suunnattuihin graafeihin, mutta tarkastelu eroaa Suomalaisen tutkimien yksinkertaisten graafien orientaatioiden tapauksesta, jotka merkitsivät aidosti vahvempaa laskennan mallia kuin porttinumerointimalli [7, s.26-27]. Modaalilogiikan näkö-

kulmasta yleistys on luonnollinen, sillä Kripke-mallien relaatioiden ei yleensä oleteta olevan symmetrisiä. Karakterisoimme vastaavanlaiset graafiongelmaluokat suunnatuille graafeille ja osoitamme näiden välillä vallitsevan hierarkian. Tuloksemme poikkeavat lähteestä [6], sillä kaikki sen tuloksista eivät yleisty suunnatuille graafeille. Erityisesti osoittautuu, että luokkien välinen järjestys ei ole lineaarinen.

1.1 Tutkielman rakenne

Luvussa 2 esittelemme modaalilogiikan, Kripke-mallin ja bisimulaation käsitteet. Tarkastelemiamme eri modaalilogiikoita on yhteensä neljä ja niiden kaavojen totuus tulkitaan Kripke-malleissa. Bisimulaatio on väline mallien väliseen vertailuun. Kaksi keskenään bisimilaarista mallia eivät erotu toisistaan maailmojen tasolla, vaan keskenään bisimilaarisissa maailmoissa ovat tosia kaikki samat modaalilogiikan kaavat. Tavanomainen bisimulaation käsite kattaa perus- ja multimodaalilogiikan, mutta tarvitsemme myös porrastetun bisimulaation käsitteen porrastettuja modaalilogiikoita varten.

Esittelemme luvussa 3 graafit, algoritmit ja graafiongelmat. Graafeja tarvitsemme abstraktien tietoverkkojen havainnollistamiseen. Hella et al. [6] tutkivat yksinkertaisia graafeja, mutta tässä tutkielmassa keskitymme suunnattuihin graafeihin. Suunnatuissa graafeissa särmät ovat nimensä mukaisesti suunnattuja, kun taas yksinkertaisissa graaifeissa niillä ei ole suuntaa. Graafiongelmat ovat graafeja koskevia ominaisuuksia, joita hajautetut algoritmit laskevat. Esittelemme luokittelun graafiongelmillä sen mukaan, kuinka vahva malli niiden ratkaisemiseen tarvitaan. Lisäksi esittelemme näitä luokkia vastaavat vakioaikaiset luokat.

Karakterisoimme vakioaikaiset graafiongelmaluokat modaalilogiikoiden avulla luvussa 4. Tätä varten näytämme myös kanonisen tavan tulkita graafi Kripke-mallina, joka riippuu tarkasteltavasta hajautetun laskennan luokasta. Malliin liitetään sitä vähemmän tietoa, mitä heikompaa luokkaa tarkastellaan. Karakterisointi tapahtuu tarkastelemalla sopivaa modaalilogiikan kieltä sopivassa mallissa.

Lopuksi luvussa 5 esitämme ja todistamme graafiongelmaluokkien hierarkiaa koskevat tulokset. Ensin osoitamme, mitkä luokista ovat identtisiä keskenään. Tämän jälkeen osoitamme, miten jäljelle jäävät luokat suhtautuvat toisiinsa. Tuloksena on täydellinen esitys luokkien välisistä inklusioista. Erityisesti osoitamme, että eräs identiteeteistä ja yksi inklusioista, jotka pätevät yksinkertaisten graafien tapauksessa, eivät päde suunnattujen graafien tapauksessa.

Lukijalta edellytämme lauselogiikan perusteiden sekä induktiotodistuksen periaatteen tunteamista. Aloitamme modaalilogiikan tarkastelun perusteista, mutta sovellamme induktioperiaatetta modaalilogiikan kaavan rakentamiseen ilman enempää perusteluita. Graafiteoriasta ja hajautetusta laskennasta emme edellytä lukijalta edeltäviä tietoja.

2 Modaalilogiikat, Kripke-mallit ja bisimulaatio

Tässä luvussa määrittelemme neljä eri modaalilogiikan kieltä ja Kripke-mallit, joissa näiden kielten kaavat tulkitaan. Lopuksi esitämme bisimulaatiot, joiden avulla voimme tunnistaa kahden Kripke-mallin pisteet, joita ei voi erottaa toisistaan annetulla modaalilogiikan kielellä. Merkitsemme jatkossa $\mathcal{P}(A) = \{B \mid B \subseteq A\}$ ja $B^A = \{f \mid f: A \rightarrow B\}$.

2.1 Modaalilogiikat ja Kripke-mallit

Aloitamme esittelemällä multi- ja perusmodaalilogiikat ja niiden kaavojen tulkitsemiseksi Kripke-mallit. Tarvitsemme propositiosymboleja $p, q, r \dots$ ja indeksejä $\alpha, \beta, \gamma \dots$ seuraavissa määritelmässä. Olkoon siis Φ joukko propositiosymboleja ja I indeksijoukko.

Määritelmä 2.1. Vrt. [6, s. 11-12]. *Multimodaalilogiikan*, MML, saamme laajentamalla lauselogiikkaa modaalioperaattoreilla $\langle \alpha \rangle$. Rakennamme multimodaalilogiikan MML(I, Φ)-kaavat seuraavasti:

$$\phi ::= q \mid \perp \mid \neg\phi \mid \phi \vee \psi \mid \langle \alpha \rangle\phi,$$

missä $q \in \Phi$ ja $\alpha \in I$.

Määritelmä 2.2. Vrt. [2, s. 9]. *Perusmodaalilogiikan*, ML, saamme multimodaalilogiikan osana rajoittamalla modaalioperaattorien lukumäärä yhteen, jota merkitsemme \diamond . Perusmodaalilogiikan ML(Φ)-kaavat rakennamme siis seuraavasti:

$$\phi ::= q \mid \perp \mid \neg\phi \mid \phi \vee \psi \mid \diamond\phi,$$

missä $q \in \Phi$.

Määritelmä 2.3. Vrt. [3, s. 4]. *Kripke-malli* on järjestetty jono $\mathcal{M} = (W, (R_\alpha)_{\alpha \in I}, \tau)$, missä W on epätyhjä joukko, $R_\alpha \subseteq W \times W$ on kaksipaikkainen joukon W relaatio jokaisella $\alpha \in I$ ja $\tau: \Phi \rightarrow \mathcal{P}(W)$ on kuvaus, joka liittää jokaiseen propositiosymboliin $q \in \Phi$ joukon W osajoukon. Kutsumme alkioita $w \in W$ mallin \mathcal{M} *pisteiksi* ja kuvausta τ mallin \mathcal{M} *valuuatioksi*. Mallia \mathcal{M} vastaava *Kripke-kehys* on järjestetty jono $\mathcal{F} = (W, (R_\alpha)_{\alpha \in I})$.

Huomautus. Kukin Kripke-mallin relaatio R_α vastaa multimodaalilogiikan modaalioperaattoria $\langle \alpha \rangle$. Mikäli indeksijoukossa I on vain yksi alkio α , voimme merkitä ainoaa relaatiota R_α kirjoittamalla R ja mallia \mathcal{M} kirjoittamalla $\mathcal{M} = (W, R, \tau)$. Tällöin relaatio R vastaa perusmodaalilogiikan modaalioperaattoria \diamond .

Määritelmä 2.4. Vrt. [2, s. 17-18]. Olkoon $\mathcal{M} = (W, (R_\alpha)_{\alpha \in I}, \tau)$ Kripke-malli ja $w \in W$. Määrittelemme, että MML(I, Φ)-kaava ϕ on *tosi mallin \mathcal{M} pisteessä w* , jota merkitsemme $\mathcal{M}, w \models \phi$, rekursiivisesti:

- $\mathcal{M}, w \models q$, jos ja vain jos $w \in \tau(q)$,
- $\mathcal{M}, w \models \perp$ ei päde koskaan,
- $\mathcal{M}, w \models \neg\phi$, jos ja vain jos $\mathcal{M}, w \not\models \phi$,
- $\mathcal{M}, w \models \phi \vee \psi$, jos ja vain jos $\mathcal{M}, w \models \phi$ tai $\mathcal{M}, w \models \psi$,

- $\mathcal{M}, w \vDash \langle \alpha \rangle \phi$, jos ja vain jos on olemassa $w' \in W$, jolle $(w, w') \in R_\alpha$ ja $\mathcal{M}, w' \vDash \phi$.

Merkitsemme jatkossa $\|\phi\|^{\mathcal{M}} := \{w \in W \mid \mathcal{M}, w \vDash \phi\}$ kaikilla modaalilogiikan kaavoilla ϕ .

Seuraavaksi esittelemme porrastetut laajennukset perus- ja multimodaalilogiikalle ja määrittelemme niiden kaavojen semantiikat.

Määritelmä 2.5. Vrt. [5, s. 1-2]. *Porrastetun modaalilogiikan (graded modal logic), GML*, saamme lisäämällä perusmodaalilogiikan modaalioperaattoriin indeksin k . Porrastetun modaalilogiikan $GML(\Phi)$ -kaavat rakennamme seuraavasti:

$$\phi ::= q \mid \perp \mid \neg\phi \mid \phi \vee \phi \mid \diamond_{\geq k}\phi,$$

missä $q \in \Phi$ ja $k \in \mathbb{Z}_+$. $GML(\Phi)$ -kaavan $\diamond_{\geq k}\phi$ semantiikan määrittelemme seuraavasti:

- $\mathcal{M}, w \vDash \diamond_{\geq k}\phi$, jos ja vain jos $|\{w' \in W \mid (w, w') \in R \text{ ja } \mathcal{M}, w' \vDash \phi\}| \geq k$.

Huomautus. Sekä multimodaalilogiikka että porrastettu modaalilogiikka ovat perusmodaalilogiikan laajennuksia. On helppo nähdä, että kaikki $ML(\Phi)$ -kaavat voidaan tulkita logiikassa MML erikoistapauksessa, jossa $|I| = 1$. Toisaalta $ML(\Phi)$ -kaavat voidaan myös tulkita logiikassa GML , kun rajoitetaan tarkasteltavat indeksit tapaukseen $k = 1$. Käsittelemme jatkossa perusmodaalilogiikkaa porrastetun modaalilogiikan erikoistapauksena.

Määritelmä 2.6. Vrt. [6, s. 12]. *Porrastetun multimodaalilogiikan (graded multimodal logic), GMML*, saamme yhdistämällä multimodaalilogiikan ja porrastetun modaalilogiikan laajennukset. Porrastetun multimodaalilogiikan $GMML(I, \Phi)$ -kaavat rakennamme seuraavasti:

$$\phi ::= q \mid \perp \mid \neg\phi \mid \phi \vee \phi \mid \langle \alpha \rangle_{\geq k}\phi,$$

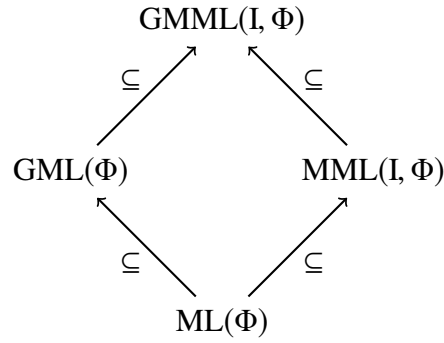
missä $q \in \Phi$, $\alpha \in I$ ja $k \in \mathbb{Z}_+$. Määrittelemme $GMML(I, \Phi)$ -kaavan $\langle \alpha \rangle_{\geq k}\phi$ semantiikan seuraavasti:

- $\mathcal{M}, w \vDash \langle \alpha \rangle_{\geq k}\phi$, jos ja vain jos $|\{w' \in W \mid (w, w') \in R_\alpha \text{ ja } \mathcal{M}, w' \vDash \phi\}| \geq k$.

Huomautus. Porrastettujen modaalilogiikoiden määritelmät voitaisiin laajentaa tapaukseen $k \in \mathbb{N}$. Tästä syntyvä modaalioperaattori olisi kuitenkin mielenkiinnoton, sillä kaava $\langle \alpha \rangle_{\geq 0}\phi$ on yhtäpitävä kaavan $\neg\perp$ kanssa kaikilla $GMML(I, \Phi)$ -kaavoilla ϕ ja $\alpha \in I$.

Huomautus. Porrastettu multimodaalilogiikka on sekä multimodaalilogiikan että porrastetun modaalilogiikan laajennus. Kaikki $MML(I, \Phi)$ -kaavat voidaan tulkita logiikassa $GMML$ rajoitumalla tapaukseen, jossa $k = 1$. Toisaalta kaikki $GML(\Phi)$ -kaavat saadaan logiikassa $GMML$ tapauksessa $|I| = 1$. Käsittelemme jatkossa porrastettua modaalilogiikkaa ja multimodaalilogiikkaa porrastetun multimodaalilogiikan erikoistapauksena.

Logiikat ML , MML , GML ja $GMML$ muodostavat siis keskenään hierarkian (Kuva 2.1). Kaikki muut logiikat ovat perusmodaalilogiikan laajennuksia, joten ML on heikoin tarkastelemistamme logiikoista. Toisaalta $GMML$ on muiden logiikoiden laajennus, tehden siitä ilmaisuvoimaltaan vahvimman. Logiikat MML ja GML ovat sen sijaan keskenään vertailemattomia. Nimittäin MML ei kykene laskemaan pisteestä lähtevien relaatioiden lukumäärää tapauksessa, jossa tarkasteltavassa mallissa on vain yksi relaatio. Toisaalta GML rajoittuu vain yhteen relaatiotransymboliin, eikä näin ollen sovellu tarkasti kuvaamaan pisteitä malleissa, joissa on monta eri relaatiota.



Kuva 2.1: Modaalilogiikoiden muodostama hierarkia.

Määrittelemme vielä modaalilogiikan kaavan modaalisen syvyyden, joka kertoo, kuinka monta sisäkkäistä modaaliooperaattoria kyseisessä kaavassa on. Lopuksi määrittelemme modaalilogiikan kaavan alikaavat.

Määritelmä 2.7. Vrt. [6, s. 12]. Olkoot ψ ja θ GMLL(I, Φ)-kaavoja. Määrittelemme GMLL(I, Φ)-kaavan ϕ modaalisen syvyyden (*modal depth*), $\text{md}(\phi)$, rekursiivisesti:

- $\text{md}(q) = 0$ kaikilla $q \in \Phi$,
- $\text{md}(\perp) = 0$,
- $\text{md}(\neg\psi) = \text{md}(\psi)$,
- $\text{md}(\psi \vee \theta) = \max\{\text{md}(\psi), \text{md}(\theta)\}$,
- $\text{md}(\langle\alpha\rangle_{\geq k}\psi) = \text{md}(\psi) + 1$, kaikilla $\alpha \in I$ ja $k \geq 0$.

Määritelmä 2.8. Vrt. [3, s. 4]. Olkoot ψ ja θ GMLL(I, Φ)-kaavoja. Määrittelemme GMLL(I, Φ)-kaavan ϕ alikaavojen (*subformula*) joukon $\text{sf}(\phi)$ rekursiivisesti:

- $\text{sf}(q) = \{q\}$ kaikilla $q \in \Phi$,
- $\text{sf}(\perp) = \{\perp\}$,
- $\text{sf}(\neg\psi) = \text{sf}(\psi) \cup \{\neg\psi\}$,
- $\text{sf}(\psi \vee \theta) = \text{sf}(\psi) \cup \text{sf}(\theta) \cup \{\psi \vee \theta\}$,
- $\text{sf}(\langle\alpha\rangle_{\geq k}\psi) = \text{sf}(\psi) \cup \{\langle\alpha\rangle_{\geq k}\psi\}$.

Kaavan ϕ alikaavat ovat siis kaavan ϕ rakennuspalikoita. Kun ϕ rakennetaan rekursiivisen määritelmän mukaisesti, niin jokaisessa vaiheessa rakentamiseen käytetyt kaavat ovat kaavan ϕ alikaavoja.

2.2 Bisimulaatiot

Seuraavaksi määrittelemme bisimulaation käsitteen. Bisimulaatiot ovat kahden Kripke-mallin välisiä relaatioita. Ne yhdistävät toisiinsa pisteitä, joilla on sama lokaali informaatio ja samat siirtymismahdollisuudet mallin sisällä. Näin kaksi bisimilaarista pistettä on mahdotonta erottaa toisistaan valitulla modaalilogiikan kielellä. Merkitsemme seuraavia määritelmiä ja lauseita varten $R_\alpha[w] = \{w' \in W \mid (w, w') \in R_\alpha\}$. Esittelemme ensimmäiseksi bisimulaation logiikalle MML.

Määritelmä 2.9. Vrt. [6, s. 12]. Olkoon Φ joukko propositiosymboleja ja I indeksijoukko. Olkoot $\mathcal{M} = (W, (R_\alpha)_{\alpha \in I}, \tau)$ ja $\mathcal{M}' = (W', (R'_\alpha)_{\alpha \in I}, \tau')$ Kripke-malleja. Relaatio $Z \subseteq W \times W'$ on *bisimulaatio*, jos se on epätyhjä ja seuraavat kolme ehtoa pätevät:

- Jos $(w, w') \in Z$ ja $q \in \Phi$, niin $w \in \tau(q)$, jos ja vain jos $w' \in \tau'(q)$.
- Jos $(w, w') \in Z$ ja $(w, v) \in R_\alpha$, niin on olemassa sellainen $v' \in W'$, että $(w', v') \in R'_\alpha$ ja $(v, v') \in Z$.
- Jos $(w, w') \in Z$ ja $(w', v') \in R'_\alpha$, niin on olemassa sellainen $v \in W$, että $(w, v) \in R_\alpha$ ja $(v, v') \in Z$.

Sanomme, että pisteet w ja w' ovat *bisimilaariset logiikassa MML*, jos on olemassa bisimulaatio Z , jolle $(w, w') \in Z$.

Logiikalle GMML bisimulaatio täytyy määritellä hieman eri tavalla, joten määrittelemme seuraavaksi porrastetun bisimulaation. Esittämämme määritelmä poikkeaa sekä de Rijkien [5] että Hellan et al. [6] esitystavoista, mutta on näiden määritelmien kanssa yhtäpitävä.

Määritelmä 2.10. Vrt. [5, s. 3]. Olkoon Φ joukko propositiosymboleja ja I indeksijoukko. Olkoot $\mathcal{M} = (W, (R_\alpha)_{\alpha \in I}, \tau)$ ja $\mathcal{M}' = (W', (R'_\alpha)_{\alpha \in I}, \tau')$ Kripke-malleja. Relaatio $Z \subseteq W \times W'$ on *porrastettu bisimulaatio*, jos se on epätyhjä ja seuraavat kaksi ehtoa pätevät:

- Jos $(w, w') \in Z$, niin $w \in \tau(q)$, jos ja vain jos $w' \in \tau'(q)$.
- Jos $(w, w') \in Z$, niin kaikilla $\alpha \in I$ on olemassa sellainen bijektio $z_\alpha : R_\alpha[w] \rightarrow R'_\alpha[w']$, että $(v, z_\alpha(v)) \in Z$ kaikilla $v \in R_\alpha[w]$.

Sanomme, että pisteet w ja w' ovat *bisimilaariset logiikassa GMML*, jos on olemassa porrastettu bisimulaatio Z , jolle $(w, w') \in Z$.

Seuraava lause toteaa bisimilaarisuuden käsitteen merkityksen: bisimilaariset pisteet toteuttavat kaikki täsmälleen samat valitun modaalilogiikan kaavat.

Lause 2.11. Vrt. [6, s. 13]. Olkoon \mathcal{L} jompikumpi logiikoista MML ja GMML, Φ joukko propositiosymboleja ja I indeksijoukko. Olkoot $\mathcal{M} = (W, (R_\alpha)_{\alpha \in I}, \tau)$ ja $\mathcal{M}' = (W', (R'_\alpha)_{\alpha \in I}, \tau')$ Kripke-malleja, $w \in W$ ja $w' \in W'$. Jos pisteet w ja w' ovat bisimilaarisia logiikassa \mathcal{L} , niin

$$\mathcal{M}, w \models \phi, \text{ jos ja vain jos } \mathcal{M}', w' \models \phi,$$

kaikilla $\mathcal{L}(I, \Phi)$ -kaavoilla ϕ .

Todistus. Vrt. [2, s. 67] ja [5, s. 3]. Todistamme lauseen induktiolla kaavan ϕ suhteen. Oletamme ensin, että \mathcal{L} on MML. Todistamme kussakin induktioaskeleessa ainoastaan implikaation vasemmalta oikealle, sillä vastakkainen suunta on samanlainen.

Jos $\phi \in \Phi$, niin oletuksesta $\mathcal{M}, w \models \phi$ seuraa, että $w \in \tau(\phi)$. Koska w ja w' ovat bisimilaarisia, niin $w' \in \tau'(\phi)$, eli $\mathcal{M}', w' \models \phi$.

Jos $\phi = \perp$, niin $\mathcal{M}, w \not\models \phi$ ja $\mathcal{M}', w' \not\models \phi$.

Teemme induktio-oletuksen, että väite ekvivalenssista pätee kaavoille ψ ja θ .

Jos $\phi = \neg\psi$ ja $\mathcal{M}, w \models \phi$, niin $\mathcal{M}, w \not\models \psi$. Nyt induktio-oletuksen perusteella $\mathcal{M}', w' \not\models \psi$ eli $\mathcal{M}', w' \models \phi$.

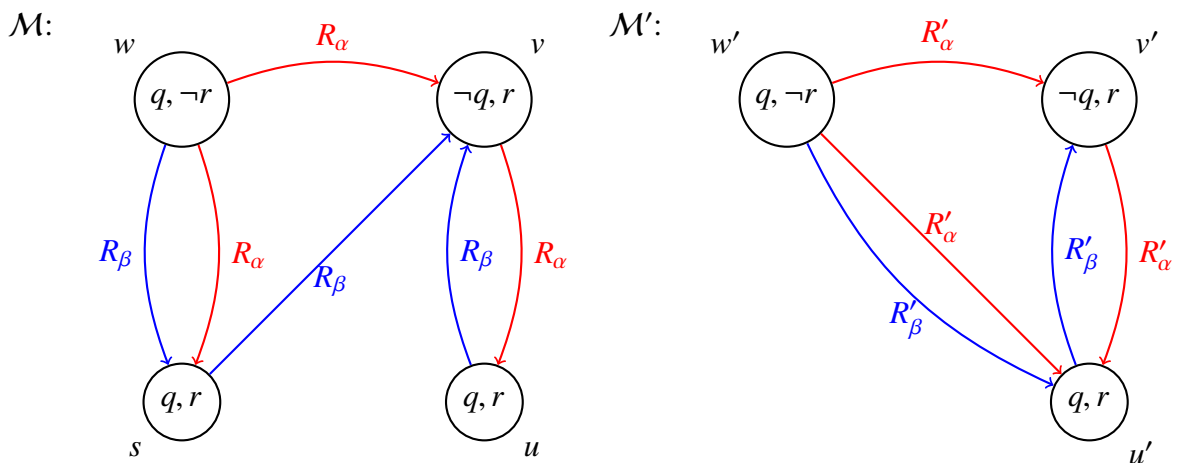
Jos $\phi = \psi \vee \theta$ ja $\mathcal{M}, w \models \phi$, niin $\mathcal{M}, w \models \psi$ tai $\mathcal{M}, w \models \theta$. Induktio-oletuksesta seuraa, että $\mathcal{M}', w' \models \psi$ tai $\mathcal{M}', w' \models \theta$, joten $\mathcal{M}', w' \models \phi$.

Jos $\phi = \langle \alpha \rangle \psi$, niin oletuksesta $\mathcal{M}, w \models \phi$ seuraa, että on olemassa sellainen $v \in W$, että $(w, v) \in R_\alpha$ ja $\mathcal{M}, v \models \psi$. Bisimulaation määritelmän nojalla on olemassa $v' \in W'$, jolle v ja v' ovat bisimilaarisia ja $(w', v') \in R'_\alpha$. Induktio-oletuksen nojalla nyt $\mathcal{M}', v' \models \psi$, joten $\mathcal{M}', w' \models \phi$.

Lause on nyt todistettu logiikalle MML. Oletetaan seuraavaksi, että \mathcal{L} on GMML. Tapaukset $\phi \in \Phi$, $\phi = \perp$, $\phi = \neg\psi$ ja $\phi = \psi \vee \theta$ todistetaan samoin kuin logiikalle MML.

Oletetaan, että väite pätee kaavalle ψ ja että $\phi = \langle \alpha \rangle_{\geq k} \psi$. Nyt oletuksesta $\mathcal{M}, w \models \phi$ seuraa, että $|X| \geq k$, missä $X = \{v \in W \mid (w, v) \in R_\alpha \text{ ja } \mathcal{M}, v \models \psi\}$. Porrastetun bisimulaation määritelmän mukaan on olemassa sellainen bijektio $z_\alpha: R_\alpha[w] \rightarrow R'_\alpha[w']$, että $(v, z_\alpha(v)) \in Z$ kaikilla $v \in X$. Siis induktio-oletuksen nojalla $\mathcal{M}', z_\alpha(v) \models \psi$ kaikilla $v \in X$. Koska z_α on injektio, niin $|\{v' \in W' \mid (w', v') \in R'_\alpha \text{ ja } \mathcal{M}', v' \models \psi\}| \geq k$. Näin ollen $\mathcal{M}', w' \models \phi$.

Toisaalta, jos $\mathcal{M}', w' \models \phi$, niin $|X'| \geq k$, missä $X' = \{v' \in W' \mid (w', v') \in R'_\alpha \text{ ja } \mathcal{M}', v' \models \psi\}$. Porrastetun bisimulaation määritelmän mukaan on olemassa bijektio $z_\alpha: R_\alpha[w] \rightarrow R'_\alpha[w']$, jolle $(v, z_\alpha(v)) \in Z$ kaikilla $v \in X$. Tämän käänteiskuvaus z_α^{-1} on myös bijektio. Siis induktio-oletuksen nojalla $\mathcal{M}, z_\alpha^{-1}(v') \models \psi$ kaikilla $v' \in X'$. Nyt koska z_α^{-1} on injektio, niin pätee $|\{v \in W \mid (w, v) \in R_\alpha \text{ ja } \mathcal{M}, v \models \psi\}| \geq k$. Näin ollen $\mathcal{M}, w \models \phi$. \square



Kuva 2.2: Kaksi keskenään bisimilaarista Kripke-mallia.

Esimerkki 2.12. Olkoon $\Phi = \{q, r\}$ ja $I = \{\alpha, \beta\}$. Olkoot $\mathcal{M} = (W, R_\alpha, R_\beta, \tau)$ ja $\mathcal{M}' = (W', R'_\alpha, R'_\beta, \tau')$ Kripke-malleja, kuten kuvassa 2.2. Vaikka mallit ovat selvästi erilaiset, on niiden välillä kuitenkin porrastettu bisimulaatio.

Määrittellemme relaation $Z = \{(w, w'), (v, v'), (u, u'), (s, u')\}$. Määritelmän 2.10 ensimmäinen ehto täyttyy, sillä pisteet w ja w' toteuttavat samat proposiiosymbolit, kuten myös pisteet v ja v' , u ja u' sekä s ja u' . Toisen ehdon osoittamista varten joudumme tarkastelemaan jokaista relaation Z paria erikseen.

Ensinnäkin parin (w, w') tapauksessa huomaamme, että $R_\alpha[w] = \{v, s\}$ ja $R'_\alpha[w'] = \{v', u'\}$. Määrittellemme siis bijektio $z_\alpha^w = \{(v, v'), (s, u')\}$. Tämä kelpaa määritelmässä vaadituksi bijektioiksi, sillä $(v, v'), (s, u') \in Z$. Samoin muodostamme bijektio $z_\beta^w = \{(s, u')\}$, joka myös toteuttaa määritelmän ehdon.

Parille (v, v') määrittellemme bijektio $z_\alpha^v = \{(u, u')\}$ ja $z_\beta^v = \emptyset$. Tyhjä kuvaus johtuu siitä, että $R_\beta[v] = \emptyset$, mutta se täyttää joka tapauksessa määritelmän ehdon.

Parille (u, u') määrittellemme bijektio $z_\alpha^u = \emptyset$ ja $z_\beta^u = \{(v, v')\}$.

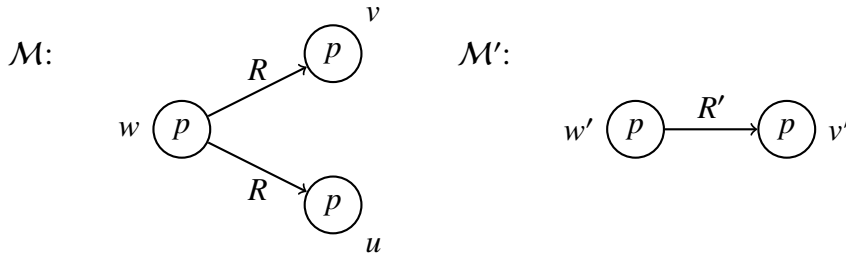
Koska $R_\alpha[s] = \emptyset = R_\alpha[u']$, $R_\beta[s] = \{v\}$ ja $R_\beta[u'] = \{v'\}$, niin parille (s, u') riittää määritellä $z_\alpha^s = z_\alpha^u$ ja $z_\beta^s = z_\beta^u$.

Koska kaikki parit on käyty läpi ja olemme löytäneet määritelmän mukaiset bijektiot, täyttyy määritelmän toinenkin ehto. Siis relaatio Z on porrastettu bisimulaatio.

Esimerkki 2.13. Olkoon $\Phi = \{p\}$ ja $|I| = 1$. Olkoot $\mathcal{M} = (W, R, \tau)$ ja $\mathcal{M}' = (W', R', \tau')$ Kripke-malleja, kuten kuvassa 2.3. Nyt $\mathcal{M}, w \models \Diamond_{\geq 2} \neg \perp$, mutta $\mathcal{M}', w' \not\models \Diamond_{\geq 2} \neg \perp$. Näin ollen lauseen 2.11 nojalla pisteiden w ja w' välillä ei ole porrastettua bisimulaatiota.

Tästä huolimatta pisteet ovat kuitenkin bisimilaarisia tavallisessa mielessä. Olkoon $Z = \{(w, w'), (v, v'), (u, v')\}$. Määritelmän 2.9 ensimmäinen ehto täyttyy triviaalisti. Toista ja kolmatta ehtoa varten joudumme tarkastelemaan ainoastaan paria (w, w') . Siirtymiä $(w, v), (w, u) \in R$ mallissa \mathcal{M} vastaa siirtymä $(w', v') \in R'$ mallissa \mathcal{M}' , joten toinen ehto täyttyy. Toisaalta siirtymää $(w', v') \in R'$ mallissa \mathcal{M}' vastaa kumpi tahansa siirtymistä $(w, v), (w, u) \in R$ mallissa \mathcal{M} . Näin ollen kolmas ehto täyttyy ja relaatio Z on bisimulaatio.

Siis logiikat ML ja MML eivät kykene erottamaan näitä malleja toisistaan, kun taas GML ja GMML kykenevät. Tämä havainnollistaa modaalilogiikoiden välisiä ilmaisuvoimaeroja ja porrastettujen modaalilogiikoiden kykyä laskea pisteestä lähtevien relaationuolten lukumääriä.



Kuva 2.3: Kaksi Kripke-mallia, joiden pisteiden w ja w' välillä on bisimulaatio. Niiden välillä ei kuitenkaan ole porrastettua bisimulaatiota.

3 Graafit, algoritmit ja graafiongelmat

Tässä luvussa esittelemme ensin suunnatut graafit ja sen jälkeen hajautetut algoritmit. Suunnatut graafit Kripke-kehysten kanssa samanlaisia struktuureja, joissa hajautetut algoritmit ajetaan. Lopuksi esittelemme graafiongelmat, joita algoritmien avulla pyritään ratkaisemaan. Lisäksi esittelemme luokittelut sekä algoritmeille että graafiongelmille.

Poikkeamme lähteestä tarkastellessamme suunnattuja graafeja. Aikaisemmin hajautettujen algoritmien yhteydessä graafien on oletettu olevan yksinkertaisia graafeja, joissa viestit kulkevat molempiin suuntiin solmujen välillä. Tämä on suunnatuille graafeille erikoistapaus, eli pyrimme yleistämään lähteen tulokset suunnatuille graafeille. Merkitsemme jatkossa $[k] = \{1, 2, \dots, k\}$ kätevyyden vuoksi.

3.1 Graafit

Määritelmä 3.1. Vrt. [8, s. 1-3]. *Suunnattu graafi* on järjestetty pari $G = (V, E)$, missä V on epätyhjä ja äärellinen joukko ja $E \subseteq V \times V$ on kaksipaikkainen joukon V relaatio. Kutsumme alkioita $v \in V$ graafin G *solmuiksi* (*vertex*) ja järjestettyjä pareja $e \in E$ graafin G *särmiksi* (*edge*). Jos $e = (u, v)$ on särmä, niin kutsumme solmua u särmän e *lähtösolmuksi* ja solmua v sen *maalisolmuksi*.

Huomautus. Suunnattu graafi $G = (V, E)$ määritellään siis samalla tavalla kuin Kripke-kehys $\mathcal{F} = (W, R)$. Struktuurit ovat siis samat, vaikka niiden merkinnät eroavat toisistaan.

Määritelmä 3.2. Vrt. [4, s. 172]. Olkoon $G = (V, E)$ suunnattu graafi. Solmun $v \in V$ *lähtöaste* (*outdegree*) on $\deg_+(v) = |\{w \in V \mid (v, w) \in E\}|$. Vastaavasti solmun v *tuloaste* (*indegree*) on $\deg_-(v) = |\{u \in V \mid (u, v) \in E\}|$.

Solmun lähtöaste siis kertoo, kuinka monta särmää solmusta lähtee ja vastaavasti tuloaste kertoo, kuinka monta särmää solmuun saapuu.

Määritelmä 3.3. Vrt. [6, s. 1]. Olkoon $G = (V, E)$ suunnattu graafi. Graafin G *lähtöaste* on $\deg_+(G) = \max\{\deg_+(v) \mid v \in V\}$ ja sen *tuloaste* on $\deg_-(G) = \max\{\deg_-(v) \mid v \in V\}$. Merkitsemme $\mathcal{F}(a, b) = \{G \mid G \text{ on graafi, } \deg_+(G) \leq a \text{ ja } \deg_-(G) \leq b\}$.

Seuraavaksi määrittelemme graafien solmuille portit, joiden kautta solmujen välinen viestintä tapahtuu. Hellan et al. [6] lähestymistavassa jokaiseen särmään liitettiin neljä porttinumeroa, sillä viestit kulkevat yksinkertaisissa graafeissa molempiin suuntiin. Sen sijaan suunnatuilla graafeilla tarvitsemme jokaiselle särmälle vain kaksi porttinumeroa.

Määritelmä 3.4. Vrt. [7, s. 3]. Olkoon $G = (V, E)$ graafi. Graafin G *lähtöportti* on järjestetty pari (v, i) , missä $v \in V$ ja $i \in [\deg_+(v)]$. Vastaavasti graafin G *tuloportti* on järjestetty pari (j, v) , missä $v \in V$ ja $j \in [\deg_-(v)]$. Merkitsemme $P_+(G) = \{(v, i) \mid v \in V \text{ ja } i \in [\deg_+(v)]\}$ ja $P_-(G) = \{(j, v) \mid v \in V \text{ ja } j \in [\deg_-(v)]\}$.

Huomautus. Jokaista graafin $G = (V, E)$ särmää vastaa täsmälleen yksi lähtöportti siinä solmussa, mistä särmä lähtee. Vastaavasti jokaista särmää vastaa täsmälleen yksi tuloportti siinä solmussa, johon särmä saapuu. Näin ollen lähtö- ja tuloportteja on yhtä monta, eli $|P_+(G)| = |P_-(G)| = |E|$.

Määritelmä 3.5. Vrt. [6, s. 1-2]. Olkoon $G = (V, E)$ graafi. Olkoon $p: P_+(G) \rightarrow P_-(G)$ bijektio. Kuvaus p on graafin G porttinumerointi, jos

$$(u, v) \in E, \text{ jos ja vain jos } p(u, i) = (j, v) \text{ joillakin } i \in [\deg_+(u)] \text{ ja } j \in [\deg_-(v)],$$

kaikilla $u, v \in V$.

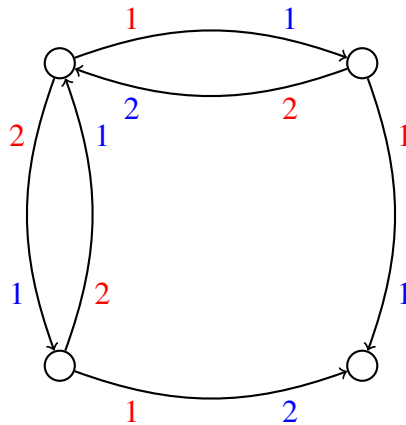
Sanomme, että porttinumerointi p on *konsistentti* (*consistent*), mikäli

$$\text{jos } p(u, i) = (j', v) \text{ ja } p(v, i') = (j, u), \text{ niin } i = j \text{ ja } i' = j',$$

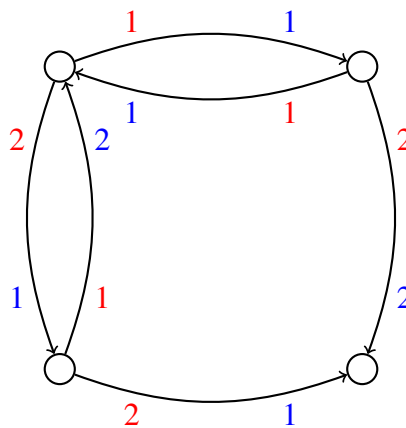
kaikilla $u, v \in V, i \in [\deg_+(u)], i' \in [\deg_+(v)], j \in [\deg_-(u)]$ ja $j' \in [\deg_-(v)]$.

Porttinumerointi siis tarkoittaa seuraavaa: Jos graafin solmu u lähettää viestin lähtöporttiin (u, i) ja $p(u, i) = (j, v)$, niin solmu v vastaanottaa kyseisen viestin tuloportista (j, v) . Konsistentissa porttinumeroinnissa, jos kahden solmun välillä kulkee särmä molempiin suuntiin, niin solmun kyseisiä särmää vastaavat lähtö- ja tuloportit saavat saman numeron. Jos kahden solmun välillä ei kulje särmää molempiin suuntiin, niin konsistentti porttinumerointi ei aseta mitään rajoituksia kyseisten solmujen välille.

Huomautus. Määritelmässä 3.5 jää auki kysymys särmien moninkertaisista numeroinneista. Ei kuitenkaan ole mahdollista, että olisi $p(u, i) = (j, v)$ ja $p(u, k) = (l, v)$, mutta $i \neq k$ tai $j \neq l$. Jos näin olisi, niin joko solmulla u ei olisi enää tarpeeksi lähtöportteja kaikille siitä lähteville särmille tai solmulla v ei olisi tarpeeksi tuloportteja kaikille siihen saapuville särmille ja näin ollen määritelmän ehto ei toteutuisi.



Kuva 3.1: Nelisolmuinen graafi ja sen porttinumerointi. Lähtöporttien numerot on merkitty punaisella ja tuloporttien sinisellä.



Kuva 3.2: Sama graafi konsistentilla porttinumeroinnilla.

3.2 Algoritmit

Seuraavaksi esittelemme hajautetut algoritmit. Ne ovat tilakoneita, jotka siirtävät solmun vaiheittain tilasta toiseen. Kussakin solmussa ajetaan samanaikaisesti kopio samasta tilakoneesta. Alussa kukin solmu saa lähtö- ja tuloasteensa perusteella alkutilan. Tämän jälkeen tilakoneiden suoritus jatkuu samanaikaisesti ja vaiheittain. Jokaisessa vaiheessa jokainen kone toteuttaa kolme asiaa:

- (i) Lähettää solmusta lähteviä särmiä pitkin viestin solmuihin, joihin kyseiset särmät saapuvat,
- (ii) ottaa solmuun saapuvia särmiä pitkin vastaan viestit solmuilta, joista kyseiset särmät lähtevät ja
- (iii) siirtää solmun seuraavaan tilaan edellisen tilan ja vastaanotettujen viestien perusteella.

Viestit lähetetään suunnattuja särmiä pitkin, eivätkä ne välttämättä kulje kumpaankin suuntaan kahden solmun välillä. Tiettyihin tiloihin saapuessaan kukin kone pysähtyy.

Määritelmä 3.6. Vrt. [6, s. 1]. *Hajautettu algoritmi* luokassa $\mathcal{F}(a, b)$ on järjestetty jono $\mathcal{A} = (Z, M, z_0, \mu, \delta)$, missä:

- Z on joukko tiloja, jonka osajoukkona on äärellinen joukko $Y \subseteq Z$ lopputiloja,
- M on joukko viestejä, joista yksi on tyhjä viesti m_0 ,
- $z_0: \{0, 1, \dots, a\} \times \{0, 1, \dots, b\} \rightarrow Z$ on kuvaus, joka antaa solmun alkutilan sen lähtö- ja tuloasteen perusteella,
- $\mu: Z \times [a] \rightarrow M$ on kuvaus, joka määrää solmun lähettämän viestin kuhunkin porttiin kussakin tilassa,
- $\delta: Z \times M^b \rightarrow Z$ on kuvaus, joka määrää solmun seuraavan tilan edeltävän tilan ja sen saamien viestien perusteella.

Lisäksi vaaditaan, että $\delta(y, \vec{m}) = y$ jokaisella $y \in Y$ ja $\vec{m} \in M^b$. Toisin sanoen lopputilaan saapunut solmu pysyy samassa tilassa ajon loppuun asti.

Huomautus. Määritelmän lopputiloja sitovan ehdon voisi ottaa lopputilojen määrääväksi ehdoksi, mutta kätevyyden vuoksi sallimme tarkasteltavien lopputilojen joukon olevan suppeampi. Emme kuitenkaan merkitse joukkoa Y algoritmin parametriksi, vaan määrittelemme sen tarpeen vaatiessa tapauskohtaisesti.

Määritelmä 3.7. Vrt. [6, s. 2-3]. Olkoon $G = (V, E) \in \mathcal{F}(a, b)$, p graafin G porttinumerointi ja $\mathcal{A} = (Z, M, z_0, \mu, \delta)$ hajautettu algoritmi luokassa $\mathcal{F}(a, b)$. Määrittelemme algoritmin \mathcal{A} ajon (G, p) :ssä rekursiivisesti:

Systeemin tila hetkellä $t \in \mathbb{N}$ on kuvaus $x_t: V \rightarrow Z$, jonka määrittelemme rekursiivisesti. Ajanhetkellä $t = 0$ määrittelemme

$$x_0(u) = z_0(\deg_+(u), \deg_-(u)),$$

kaikilla $u \in V$.

Olkoon x_t annettu ajanhetkellä t . Olkoot $u, v \in V$ ja $p(v, i) = (j, u)$ joillakin $i \in [\deg_+(v)]$ ja $j \in [\deg_-(u)]$. Solmun u vastaanottaman viestin tuloportista (j, u) ajanhetkellä $t + 1$ määrittelemme seuraavasti:

$$h_{t+1}(j, u) = \mu(x_t(v), i) \in M.$$

Siis u vastaanottaa ajanhetkellä $t + 1$ saman viestin, jonka v lähettää porttiin (v, i) samalla ajanhetkellä.

Määrittelemme vektorin kaikkien solmun u ajanhetkellä $t + 1$ vastaanottamien viestien varastointiin seuraavasti:

$$\vec{h}_{t+1}(u) = (h_{t+1}(1, u), h_{t+1}(2, u), \dots, h_{t+1}(\deg_-(u), u), m_0, m_0, \dots, m_0) \in M^b.$$

Nyt solmun $u \in V$ tilan ajanhetkellä $t + 1$ määrittelemme seuraavasti:

$$x_{t+1}(u) = \delta(x_t(u), \vec{h}_{t+1}(u)).$$

Sanomme hajautetun algoritmin \mathcal{A} pysähtyvän ajassa $T(G, p)$:ssä, jos kaikilla $u \in V$ pätee $x_T(u) \in Y$.

Sanomme hajautetun algoritmin \mathcal{A} pysähtyvän ajanhetkellä $T(G, p)$:ssä, jos kaikilla $u \in V$ pätee $x_T(u) \in Y$ ja kaikilla $T' < T$ pätee $x_{T'}(v) \notin Y$ jollakin $v \in V$.

Kummassakin tapauksessa sanomme, että $S_{\mathcal{A},(G,p)} = x_T$ on hajautetun algoritmin \mathcal{A} tuloste (G, p) :ssä ja $S_{\mathcal{A},(G,p)}(u) = x_T(u)$ on solmun $u \in V$ lokaali tuloste.

Huomautus. Vektorissa $\vec{h}_{t+1}(u)$ tyhjiä viestejä m_0 on lopussa niin monta kuin tarvitaan, jotta $\vec{h}_{t+1}(u) \in M^b$. Lukumäärä voi vaihdella kokonaislukuvälillä $[0, b]$. Todellisuudessa mikään muu solmu ei lähetä näitä tyhjiä viestejä, jotka eivät sisällä informaatiota; merkitsemme näin vain kätevyyden vuoksi.

3.3 Algoritmien luokat

Seuraavaksi määrittelemme algoritmien luokat sen mukaan, miten solmut välittävät viestejä keskenään. Riippuen tarkasteltavasta algoritmista voi olla, etteivät solmut tiedä mikä viesti vastaanotettiin mistäkin tuloportista tai kuinka monta identtistä viestiä vastaanotettiin. Toisaalta viestejä lähettäessään solmut voivat lähettää aina saman viestin kaikkiin lähtöportteihinsa.

Olkoon $\mathcal{A} = (Z, M, z_0, \mu, \delta)$ hajautettu algoritmi luokassa $\mathcal{F}(a, b)$ ja $\vec{h} = (h_1, \dots, h_b) \in M^b$ vektori viestejä. Määrittelemme kuvaukset $s: M^b \rightarrow \mathcal{P}(M)$ ja $ms: M^b \rightarrow \mathcal{P}(M \times [b])$ seuraavasti:

- $s(\vec{h}) = \{h_1, \dots, h_b\}$,
- $ms(\vec{h}) = \{(h_i, n) \mid i \in [b], n = |\{j \in [b] \mid h_j = h_i\}|\}$.

Kuvaus s siis kuvaa vektorin sitä vastaavaksi joukoksi (set) ja kuvaus ms kuvaa sen multijoukolle (multiset), eli joukolle, jossa viestien moninkertaisuudet säilyvät. Multijoukkoon siirtyessä siis katoaa indeksöity tieto, mistä solmusta kukin viesti tulee ja joukkoon siirtyessä katoaa myös tieto siitä, kuinka monta identtistä viestiä solmu vastaanottaa.

Määritelmä 3.8. Vrt. [6, s. 4]. Olkoon $\mathcal{A} = (Z, M, z_0, \mu, \delta)$ hajautettu algoritmi luokassa $\mathcal{F}(a, b)$. Määrittelemme luokat **V** (Vector), **M** (Multiset), **S** (Set) ja **B** (Broadcast) seuraavasti:

- $\mathcal{A} \in \mathbf{V}$, aina,
- $\mathcal{A} \in \mathbf{M}$, mikäli jos $ms(\vec{h}) = ms(\vec{h}')$, niin $\delta(z, \vec{h}) = \delta(z, \vec{h}')$ kaikilla $z \in Z$,
- $\mathcal{A} \in \mathbf{S}$, mikäli jos $s(\vec{h}) = s(\vec{h}')$, niin $\delta(z, \vec{h}) = \delta(z, \vec{h}')$ kaikilla $z \in Z$,

- $\mathcal{A} \in \mathbf{B}$, mikäli $\mu(z, i) = \mu(z, i')$ kaikilla $z \in Z$ ja $i, i' \in [a]$.

On helppo nähdä, että $\mathbf{S} \subseteq \mathbf{M} \subseteq \mathbf{V}$. Luokan \mathbf{V} algoritmeissa solmut vastaanottavat viestinsä vektoreina. Luokassa \mathbf{M} solmut sen sijaan vastaanottavat viestinsä multijoukkoina ja luokassa \mathbf{S} ne vastaanottavat viestinsä joukkoina. Luokkien \mathbf{M} ja \mathbf{S} algoritmeissa solmuilla ei siis ole tietoa, mistä tuloportista kukin viesti tuli, joten näiden luokkien yhteydessä konsistenssi porttinumeroinnissa ei merkitse mitään.

On ilmeistä, että $\mathbf{B} \subseteq \mathbf{V}$. Luokan \mathbf{V} algoritmeissa solmut voivat lähettää eri viestit eri särmiä pitkin samassa laskennan vaiheessa. Luokassa \mathbf{B} solmut sen sijaan lähettävät saman viestin kaikkia särmiä pitkin kussakin vaiheessa. Luokassa \mathbf{B} solmut eivät täten tiedä, mistä lähtöportista kukin viesti lähetetään, joten tämänkin luokan yhteydessä konsistenssi porttinumeroinnissa ei merkitse mitään.

3.4 Graafiongelmat

Seuraavaksi määrittelemme graafiongelmat ja esittelemme niiden keskeisen luokittelun.

Määritelmä 3.9. Vrt. [6, s. 3]. *Graafiongelma* on kuvaus Π , joka jollakin Y kuvaa jokaiselle suunnatulle graafille $G = (V, E)$ joukon *ratkaisuja* $\Pi(G) \in \mathcal{P}(Y^V)$. Jokainen ratkaisu on siis kuvaus $V \rightarrow Y$.

Määritelmä 3.10. Vrt. [6, s. 3]. Olkoon \mathcal{A} hajautettu algoritmi luokassa $\mathcal{F}(a, b)$ ja Π graafiongelma. Sanomme, että \mathcal{A} *ratkaisee graafiongelman* Π *luokassa* $\mathcal{F}(a, b)$, jos kaikilla graafeilla $G = (V, E) \in \mathcal{F}(a, b)$ ja niiden porttinumeroinneilla p pätee $S_{\mathcal{A},(G,p)} \in \Pi(G)$. Toisin sanoen algoritmin tuloste (G, p) :ssä kuuluu ratkaisujoukkoon $\Pi(G)$.

Määritelmä 3.11. Vrt. [6, s. 3]. Olkoon $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\}$ perhe hajautettuja algoritmeja ja Π graafiongelma. Sanomme, että \mathcal{A} *ratkaisee graafiongelman* Π , jos $\mathcal{A}_{a,b}$ ratkaisee graafiongelman Π luokassa $\mathcal{F}(a, b)$ kaikilla $a, b \in \mathbb{Z}_+$.

Määritelmä 3.12. Vrt. [6, s. 3]. Olkoon Π graafiongelma ja $T: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ kuvaus. Olkoon $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\}$ perhe hajautettuja algoritmeja. Sanomme, että \mathcal{A} *ratkaisee graafiongelman* Π *ajassa* T , jos kaikilla $a, b \in \mathbb{Z}_+$, graafeilla $G = (V, E) \in \mathcal{F}(a, b)$ ja niiden porttinumeroinneilla p pätevät seuraavat ehdot:

- $\mathcal{A}_{a,b}$ pysähtyy ajassa $T(a, b, |V|)$ (G, p) :ssä.
- $S_{\mathcal{A}_{a,b},(G,p)} \in \Pi(G)$.

Lisäksi sanomme, että \mathcal{A} *ratkaisee graafiongelman* Π *ajassa* T *olettaen konsistenssin*, jos \mathcal{A} ratkaisee graafiongelman Π ajassa T kaikilla konsistenteilla porttinumeroinneilla p .

Sanomme, että \mathcal{A} *ratkaisee graafiongelman* Π *vakioajassa* T (\mathcal{A} on *graafiongelman* Π *lokaali algoritmi*), jos \mathcal{A} on graafiongelman Π algoritmi ja on olemassa $T': \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, jolle $T(a, b, n) = T'(a, b)$ kaikilla $n \in \mathbb{N}$. Toisin sanoen lokaalin algoritmin ajoon kuluva aika ei riipu siitä, kuinka monta solmua graafissa on.

Huomautus. Jokainen perhe hajautettuja algoritmeja $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\}$ määrittelee graafiongelman $\Pi_{\mathcal{A}}$. Jos G on graafi, niin $\Pi_{\mathcal{A}}$ kuvaa sen joukolle kuvauksia, jotka ovat täsmälleen hajautettujen algoritmien $\mathcal{A}_{a,b}$ tulosteet (G, p) :ssä, missä $a, b \in \mathbb{Z}_+$ ja p on graafin G porttinumerointi.

Seuraavaksi esitämme esimerkin graafiongelmasta. Käytämme esimerkissä hieman vahvempia oletuksia kuin tähän mennessä, mutta esittelemämme määritelmät yleistyvät luonnollisesti tähän tapaukseen.

Esimerkki 3.13. Tarvitsemme tätä esimerkkiä varten muutamia uusia käsitteitä. Sanomme suunnatun graafin olevan *yhtenäinen*, mikäli sitä vastaava yksinkertainen graafi on yhtenäinen (ks. yhtenäiset graafit [8, s. 49]).

Sanomme, että suunnatun graafin $G = (V, E)$ *polku* on jono $(v_0, e_1, v_1, e_2, \dots, e_n, v_n)$, jolle pätevät seuraavat ehdot:

- $v_i \in V$ kaikilla $i \in \{0, 1, \dots, n\}$.
- $e_i \in E$ kaikilla $i \in [n]$.
- $e_i = (v_{i-1}, v_i)$ kaikilla $i \in [n]$.

Kyseinen polku on *suora*, mikäli $v_i \neq v_j$ ja $e_i \neq e_j$ kaikilla $i \neq j$. Tällöin sanomme, että kyseisen suoran polun *pituus* on n (vrt. [4, s. 12]).

Määrittelemme, että yhtenäinen suunnattu graafi $G = (V, E)$ on (*suunnattu*) *puu*, mikäli seuraavat kaksi ehtoa pätevät:

- On olemassa yksikäsitteinen $v \in V$, jolle $\deg_+(v) = 0$.
- Kaikilla $u \in V \setminus \{v\}$ pätee, että $\deg_+(u) = 1$.

Solmua v kutsumme puun *juureksi*. On tarpeellista, että tarkasteltava graafi on yhtenäinen, sillä muuten yllä olevat kaksi ehtoa eivät riitä kytkemään kaikkia graafin solmuja juureen.

Rajoitumme tässä esimerkissä tarkastelemaan yhtenäisiä suunnattuja graafeja. Oletamme myös, että tarkasteltavissa algoritmeissa syötegraafin solmuilla on käytössään tieto siitä, mikä on graafin pisimmän suoran polun pituus sekä laskuri, jolla ne tietävät, missä laskennan vaiheessa ajo milläkin hetkellä on. Tätä varten määrittelemme graafien luokat $\mathcal{F}(a, b, l)$, missä $G \in \mathcal{F}(a, b, l)$, jos ja vain jos $\deg_+(G) \leq a$, $\deg_-(G) \leq b$ ja graafin G pisimmän suoran polun pituus on enintään l .

Tarkastelemme graafiongelmaa Π , jossa tarkoituksena on selvittää, onko syötegraafi puu. Olkoon siis $G = (V, E) \in \mathcal{F}(a, b, l)$. Luomme tarkastelua varten hajautetun algoritmin $\mathcal{A}_{a,b,l} = (Z, M, z_0, \mu, \delta)$.

Olkoon $Z = \{z_t \mid t \in [l + 1]\} \cup \{0, 1\}$, missä $Y_0 = \{0\}$ ja $Y_1 = \{1\}$. Tässä tilat z_t ovat väliaikaisia tiloja solmuille, tila 0 on kieltävä lopputila ja 1 on myönteinen lopputila. Jos ajon lopussa yksikin solmu on tilassa 0, niin graafi ei ole puu. Mikäli sen sijaan kaikki solmut päätyvät tilaan 1, niin kyseessä on puu.

Olkoon $M = Z \cup \{m_0\}$. Toisin sanoen graafin solmujen toisilleen lähettämät viestit ovat yksinkertaisesti tiloja. Solmu lähettää särmiään pitkin ajanhetkellä t viestin $z \in Z$, mikäli se on hetkellä t tilassa z . Solmut siis kommunikoivat toisilleen lähettämällä viestinä senhetkisen tilansa.

Määrittelemme kuvauksen $z_0: \{0, 1, \dots, a\} \times \{0, 1, \dots, b\} \rightarrow Z$ seuraavasti:

$$z_0(i, j) = \begin{cases} 0, & \text{kun } i \geq 2 \\ 1, & \text{kun } j = 0 \\ z_1, & \text{muulloin} \end{cases}$$

Siis lähtöasteen ollessa 2 tai enemmän, solmu siirtyy tilaan 0, sillä kyseessä ei voi olla puu. Tuloasteen ollessa 0, solmu siirtyy tilaan 1, sillä puun haarojen päät eivät tarvitse tai edes voi saada enempää informaatiota. Muussa tapauksessa solmut aloittavat tilasta z_1 .

Kuvauksen $\mu: Z \times [a] \rightarrow M$ määrittelemme yksinkertaisesti siten, että $\mu(z, i) = z$ kaikilla $z \in Z$ ja $i \in [a]$. Solmut lähettävät siis viesteinä vain tiedon senhetkisestä tilastaan.

Lopuksi määrittelemme kuvauksen $\delta: Z \times M^b \rightarrow Z$ seuraavasti:

$$\delta(z, (m_1, \dots, m_b)) = \begin{cases} 0, & \text{kun } m_j = 0 \text{ jollakin } j \in [b] \text{ tai } z = z_{l+1} \text{ tai } z = 0 \\ 1, & \text{kun } m_j \in \{1, m_0\} \text{ kaikilla } j \in [b] \text{ tai } z = 1 \\ z_{t+1}, & \text{kun } z = z_t, t \leq l, m_j \neq 0 \text{ kaikilla } j \in [b] \\ & \text{ja on olemassa } j' \in [b], \text{ jolle } m_{j'} \notin \{1, m_0\} \end{cases}$$

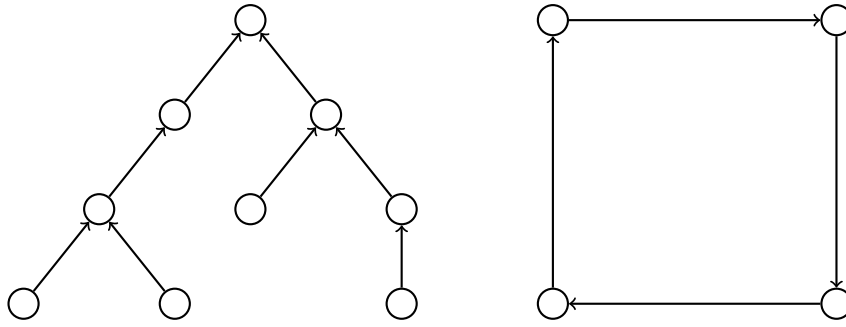
missä $z \in Z$ ja $m_j \in M$ kaikilla $j \in [b]$. Siis jos jokin solmun vastaanottamista viesteistä kertoo, että viestin lähettänyt solmu on jo päätyneet hylkäävään tilaan, tekee viestin vastaanottava solmu samoin. Jos taas kaikki solmun vastaanottamat viestit kertovat, että viestit lähettäneet solmut ovat päätyneet hyväksyvään tilaan, tekee viestit vastaanottava solmu samoin. Lopputilaan päätyneet solmut pysyvät samassa tilassa. Mikäli solmu ei ole lopputilassa ja sen saamat viestit eivät vie sitä lopputilaan, niin solmu siirtyy järjestyksessä tilasta z_t tilaan z_{t+1} , kunnes se päätyy tilaan z_{l+1} , josta se siirtyy lopputilaan 0.

Algoritmin idea on, että kukin solmu pitää kirjaa sen viereisistä solmuista. Puun haarojen päissä olevat solmut eivät saa viestejä ja päätyvät tilaan 1. Viestit kulkevat puun haarojen päistä sen juurta kohti. Viimeisenä lopputilaan pääsee juuri, mikä päättää laskennan. Mikäli tarkasteltava graafi ei ole puu, niin laskuri, eli tilat z_1, z_2, \dots, z_{l+1} , pitää huolen siitä, että laskenta pysähtyy, oli syötegraafi millainen tahansa. Tämä johtuu siitä, että mikäli graafi olisi puu, niin algoritmin ajo päättyisi vääjäämättä $l + 1$:n askeleen jälkeen, koska l on pisin mahdollinen matka puun haaran päästä sen juureen.

Nyt voimme sanoa analogisesti määritelmän 3.12 kanssa, että $\mathcal{A} = \{\mathcal{A}_{a,b,l} \mid a, b, l \in \mathbb{Z}_+\}$ on graafiongelman Π algoritmi. Sitä emme kuitenkaan voi kutsua lokaaliksi algoritmiksi, sillä algoritmin ajonaika riippuu graafin solmujen lukumäärästä.

Tässä esimerkissä syötegraafin lähtö- ja tuloasteella on lopputuloksen kannalta eri merkitys. Graafin tuloaste $\deg_-(G)$ saa vapaasti vaihdella. Itse asiassa jos graafin solmujen lukumäärä on vakio ja graafi todella on puu, niin suurempi tuloaste johtaa nopeampaan laskentaan. Graafin lähtöaste $\deg_+(G)$ sen sijaan vaikuttaa merkittävämmiin laskennan tulokseen. Mikäli $\deg_+(G) > 1$, niin kyseinen graafi ei voi olla puu.

Kuvan 3.3 vasemmanpuoleinen graafi osoittautuu puuksi, kun siinä ajetaan hajautettu algoritmi $\mathcal{A}_{1,2,3}$. Laskenta päättyy ajanhetkellä 3. Ilman laskuria kuvan 3.3 oikeanpuoleista graafia ei voisi tarkastella, sillä algoritmin ajo jatkuisi siinä loputtomiin. Laskurin avulla algoritmi $\mathcal{A}_{1,1,3}$ osoittaa ajanhetkellä 4, että se ei ole puu.



Kuva 3.3: Kuvassa vasemmalla puu ja oikealla nelisolmuinen suunnattu sykli. Kummankin graafin pisin suora polku on pituudeltaan 3.

3.5 Graafiongelmien luokat

Luvun lopuksi määrittelemme nyt graafiongelmien luokat algoritmien luokkien avulla.

Määritelmä 3.14. Vrt. [6, s. 5]. Olkoon Π graafiongelma. Määrittelemme luokat VV_c , VV , MV , SV , VB , MB ja SB seuraavasti:

- (i) $\Pi \in VV_c$, jos on olemassa hajautettujen algoritmien perhe \mathcal{A} , joka ratkaisee graafiongelman Π olettaen konsistenssin ja $\mathcal{A} \in \mathbf{V}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (ii) $\Pi \in VV$, jos on olemassa graafiongelman Π algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{V}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (iii) $\Pi \in MV$, jos on olemassa graafiongelman Π algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{M}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (iv) $\Pi \in SV$, jos on olemassa graafiongelman Π algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{S}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (v) $\Pi \in VB$, jos on olemassa graafiongelman Π algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{B}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (vi) $\Pi \in MB$, jos on olemassa graafiongelman Π algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{M} \cap \mathbf{B}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (vii) $\Pi \in SB$, jos on olemassa graafiongelman Π algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{S} \cap \mathbf{B}$ kaikilla $\mathcal{A} \in \mathcal{A}$.

Vastaavasti määrittelemme luokat vakioaikaisille luokille.

Määritelmä 3.15. Vrt. [6, s. 5]. Olkoon Π graafiongelma. Määrittelemme luokat $VV_c(1)$, $VV(1)$, $MV(1)$, $SV(1)$, $VB(1)$, $MB(1)$ ja $SB(1)$ seuraavasti:

- (i) $\Pi \in VV_c(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , joka ratkaisee graafiongelman Π olettaen konsistenssin ja $\mathcal{A} \in \mathbf{V}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (ii) $\Pi \in VV(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{V}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (iii) $\Pi \in MV(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{M}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (iv) $\Pi \in SV(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{S}$ kaikilla $\mathcal{A} \in \mathcal{A}$,

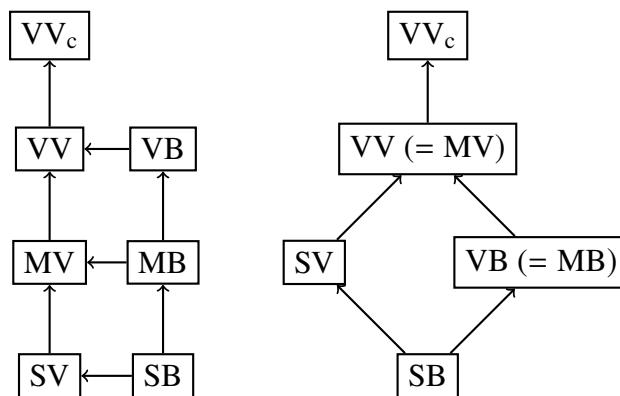
- (v) $\Pi \in \mathbf{VB}(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{B}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (vi) $\Pi \in \mathbf{MB}(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{M} \cap \mathbf{B}$ kaikilla $\mathcal{A} \in \mathcal{A}$,
- (vii) $\Pi \in \mathbf{SB}(1)$, jos on olemassa graafiongelman Π lokaali algoritmi \mathcal{A} , jolle $\mathcal{A} \in \mathbf{S} \cap \mathbf{B}$ kaikilla $\mathcal{A} \in \mathcal{A}$.

Nämä graafiongelmiä luokat ovat tämän tutkielman keskeisin aihe. On helppoa nähdä algoritmien luokittelun perusteella, että eräät luokat ovat toisten luokkien osajoukkoja, kuten $\mathbf{SV} \subseteq \mathbf{MV} \subseteq \mathbf{VV} \subseteq \mathbf{VV}_c$ ja $\mathbf{SB} \subseteq \mathbf{MB} \subseteq \mathbf{VB}$. Kuitenkin esimerkiksi luokkien \mathbf{SV} ja \mathbf{VB} välillä ei näy selvää yhteyttä, kun taas jotkin yhteydet voivatkin olla vahvempia, esimerkiksi jos osoittautuisi, että $\mathbf{MV} = \mathbf{VV}$. Lähteessä [6] osoittautui, että yksinkertaisten graafien tapauksessa luokat muodostavat keskenään lineaarijärjestyksen. Suunnattujen graafien tapauksessa tämä tulos ei kuitenkaan enää päde. Toisin kuin artikkelissa, luokat \mathbf{MV} ja \mathbf{SV} osoittautuvat eri luokiksi ja luokat \mathbf{SV} ja \mathbf{VB} ovat keskenään vertailemattomia. Luokkien muodostamasta järjestyksestä löytyy kuitenkin kaksi ketjua, jotka yhdessä näyttävät kaikki luokkien väliset yhteydet:

$$\begin{aligned} \mathbf{SB} \subsetneq \mathbf{MB} = \mathbf{VB} \subsetneq \mathbf{MV} = \mathbf{VV} \subsetneq \mathbf{VV}_c \\ \mathbf{SB} \subsetneq \mathbf{SV} \subsetneq \mathbf{MV} = \mathbf{VV} \subsetneq \mathbf{VV}_c. \end{aligned}$$

Vakioaikaisille luokille saamme vastaavat ketjut:

$$\begin{aligned} \mathbf{SB}(1) \subsetneq \mathbf{MB}(1) = \mathbf{VB}(1) \subsetneq \mathbf{MV}(1) = \mathbf{VV}(1) \subsetneq \mathbf{VV}_c(1) \\ \mathbf{SB}(1) \subsetneq \mathbf{SV}(1) \subsetneq \mathbf{MV}(1) = \mathbf{VV}(1) \subsetneq \mathbf{VV}_c(1). \end{aligned}$$



Kuva 3.4: Vasemmalla graafiongelmaluokkien väliset triviaalit osajoukkorelaatiot. Oikealla kaikki luokkien väliset osajoukkorelaatiot.

Osoitamme näitä tuloksia ennen luvussa 4, että jokainen vakioaikaisista luokista voidaan karakterisoida sopivan modaaliologiikan avulla, mikä mahdollistaa bisimulaation käytön tulosten todistamisessa.

4 Vakioaikaisten luokkien karakterisointi modaalilogiikoilla

Tässä luvussa esittelemme ja todistamme esiteltyjen modaalilogiikoiden ja vakioaikaisten luokkien välisen yhteyden. Jatkoa varten sovimme, että joukkojen A ja B leikkauksen ollessa tyhjä merkitsemme niiden yhdistettä $A \uplus B$.

Mainitsimme aikaisemmin, että graafit $G = (V, E) \in \mathcal{F}(a, b)$ ovat verrattavissa Kripke-kehyksiin $\mathcal{F} = (W, R)$, missä määrittelemme, että $W = V$ ja $R = E^{-1}$. Siis siirtyessämme graafeista kehyksiin tulkitsemme solmut pisteinä ja särmät vastakkaiseen suuntaan kulkevana relaationa. Tämä käänös johtuu siitä, että solmut lähettävät viestinsä särmien suuntaisesti. Malleissa sen sijaan pisteet saavat tiedon kaavan $\diamond\phi$ totuudesta mallin relaation avulla, jolloin tieto tulkitaan siinä pisteessä, josta relaatio lähtee.

Samalla tavoin jos p on graafin G porttinumerointi, niin saamme muodostettua struktuuria (G, p) vastaavan Kripke-mallin $\mathcal{M}(G, p) = (W, (R_\alpha)_{\alpha \in I}, \tau)$. Itse asiassa näitä malleja on neljä, riippuen siitä, minkä luokan algoritmeja tarkastelemme. Aloitamme yleisestä tapauksesta. Tätä varten tarvitsemme joukon propositiosymboleja ja indeksijoukon. Määrittelemme $\Phi_{a,b} = \{q_{i,j} \mid i \in \{0, 1, \dots, a\}, j \in \{0, 1, \dots, b\}\}$ ja $I = \{(i, j) \mid i \in [a], j \in [b]\}$. Määrittelemme jälleen, että $W = V$. Tämän lisäksi määrittelemme relaatiot $R_{(i,j)}$ seuraavasti:

$$R_{(i,j)} = \{(u, v) \in V \times V \mid p(v, i) = (j, u)\},$$

kaikilla $i \in [a]$ ja $j \in [b]$. Lopuksi määrittelemme vielä valuaation $\tau: \Phi_{a,b} \rightarrow \mathcal{P}(V)$ seuraavasti:

$$\tau(q_{i,j}) = \{v \in V \mid \deg_+(v) = i, \deg_-(v) = j\},$$

kaikilla $i \in \{0, 1, \dots, a\}$ ja $j \in \{0, 1, \dots, b\}$. Siis kuten kehysten tapauksessa, solmut tulkitaan pisteinä ja särmät kääntävät suuntaa. Lisäksi särmät jaetaan useaksi relaatioksi porttinumeroinnin avulla. Lopuksi propositiosymbolin totuus solmussa (pisteessä) määräytyy solmun lähtö- ja tuloasteen mukaan.

Huomautus. Yllä on käytetty muuttujia i ja j kahdessa merkityksessä: pisteiden porttinumeroina sekä pisteiden lähtö- ja tuloasteina. Porttinumerointi alkaa numerosta 1, joten indeksien tapauksessa i ja j eivät saa arvoikseen 0. Lähtö- ja tuloasteen tapauksessa sen sijaan on mahdollista, että graafissa olisi solmuja, joista ei lähde särmää tai joihin niitä ei tule. Jos solmusta ei lähde särmää, se ei kykene lähettämään viestejä muihin solmuihin ja käyttäytyy ikään kuin tarkkailijana. Jos taas solmuun ei saavu särmää, niin se ei kykene vastaanottamaan viestejä, mutta voi silti lähettää niitä ja vaikuttaa laskennan tulokseen.

Jos Π on graafiongelma ja \mathcal{A} on sen lokaali algoritmi, niin algoritmin \mathcal{A} ajoa (G, p) :ssä vastaa jokin joukko porrastetun multimodaalilogiikan kaavoja $\psi \in \text{GMML}(I, \Phi_{a,b})$. Riippuen siitä, mihin luokkaan algoritmi \mathcal{A} sisältyy, voivat kaavat ψ kuulua myös johonkin heikommista modaalilogiikoista. Vastaavasti jos Ψ on porrastetun multimodaalilogiikan kaavojen $\psi \in \text{GMML}(I, \Phi_{a,b})$ joukko, niin kaavojen totuus mallissa $\mathcal{M}(G, p) = (W, (R_\alpha)_{\alpha \in I}, \tau)$ voidaan määrittää jollakin lokaalilla algoritmilla \mathcal{A} (joka siis ratkaisee jonkin graafiongelman $\Pi_{\mathcal{A}}$ vakioajassa). Rajoitumme tässä tarkastelemaan algoritmeja, joita vastaa yksi modaalilogiikan kaava, eli algoritmeja, joiden lopputilojen joukko on yhdiste $Y = Y_0 \uplus Y_1$. Joukon Y_0 lopputilat siis vastaavat kyseisen kaavan epätotuutta ja joukon Y_1 lopputilat kaavan totuutta annetussa pisteessä.

Algoritmin \mathcal{A} ajo (G, p) :ssä samastetaan modaalilogiikkaan seuraavasti: Alussa jokaisessa pisteessä tiedetään tosiksi vain niissä toteutuvat propositiosymbolit (sekä kaikki pisteessä todet

lauselogiikan kaavat, joita kaikista proposiiosymboleista ja konnektiiveista voi muodostaa). Tämän jälkeen solmujen lähettämät ja vastaanottamat viestit tulkitaan modaalioperaattorien avulla seuraavasti:

$\mathcal{M}(G, p), v \vDash \langle \alpha \rangle \psi$, jos ja vain jos solmu v vastaanottaa viestin " ψ on tosi" joltakin solmulta u , jolle $(v, u) \in R_\alpha$.

Siis ajanhetkellä T kukin solmu saa tietoonsa kaikki kaavat, jotka ovat siinä tosia ja joiden modaalinen syvyys on täsmälleen T . Ajon aikana solmu saa joka kierroksella aina modaalisyvydeltään yhtä suurempien kaavojen totuusarvot. Solmun siirtyessä tilasta toiseen on kyse siitä, että jonkin valitun kaavan totuus kyseisessä solmussa on joko saatu vahvistettua tai kumottua.

Tarvitsemme neljää eri perhettä relaatioita karakterisoidaksemme jokaisen eri luokan algoritmeja. Tähän mennessä määrittelemämme perhe $(R_{(i,j)})_{(i,j) \in I}$ sopii tapaukseen, jossa solmuilla on tieto lähtö- ja tuloporttinumeroistaan. Muita tapauksia varten määrittelemme seuraavat kolme variaatiota:

- $R_{(i,*)} = \bigcup_{j \in [b]} R_{(i,j)}$ kaikilla $i \in [a]$,
- $R_{(*,j)} = \bigcup_{i \in [a]} R_{(i,j)}$ kaikilla $j \in [b]$,
- $R_{(*,*)} = \bigcup_{(i,j) \in I} R_{(i,j)}$.

Kukin neljästä perheestä tulee lisäksi tarvitsemaan oman indeksijoukon.

Nyt määrittelemme mallin $\mathcal{M}(G, p)$ neljä eri versiota seuraavasti:

- $\mathcal{M}_{a,b}(G, p) = (W, (R_\alpha)_{\alpha \in I_{a,b}}, \tau)$, missä $I_{a,b} = [a] \times [b]$,
- $\mathcal{M}_{a,*}(G, p) = (W, (R_\alpha)_{\alpha \in I_{a,*}}, \tau)$, missä $I_{a,*} = [a] \times \{*\}$,
- $\mathcal{M}_{*,b}(G, p) = (W, (R_\alpha)_{\alpha \in I_{*,b}}, \tau)$, missä $I_{*,b} = \{*\} \times [b]$,
- $\mathcal{M}_{*,*}(G, p) = (W, (R_\alpha)_{\alpha \in I_{*,*}}, \tau)$, missä $I_{*,*} = \{(*, *)\}$.

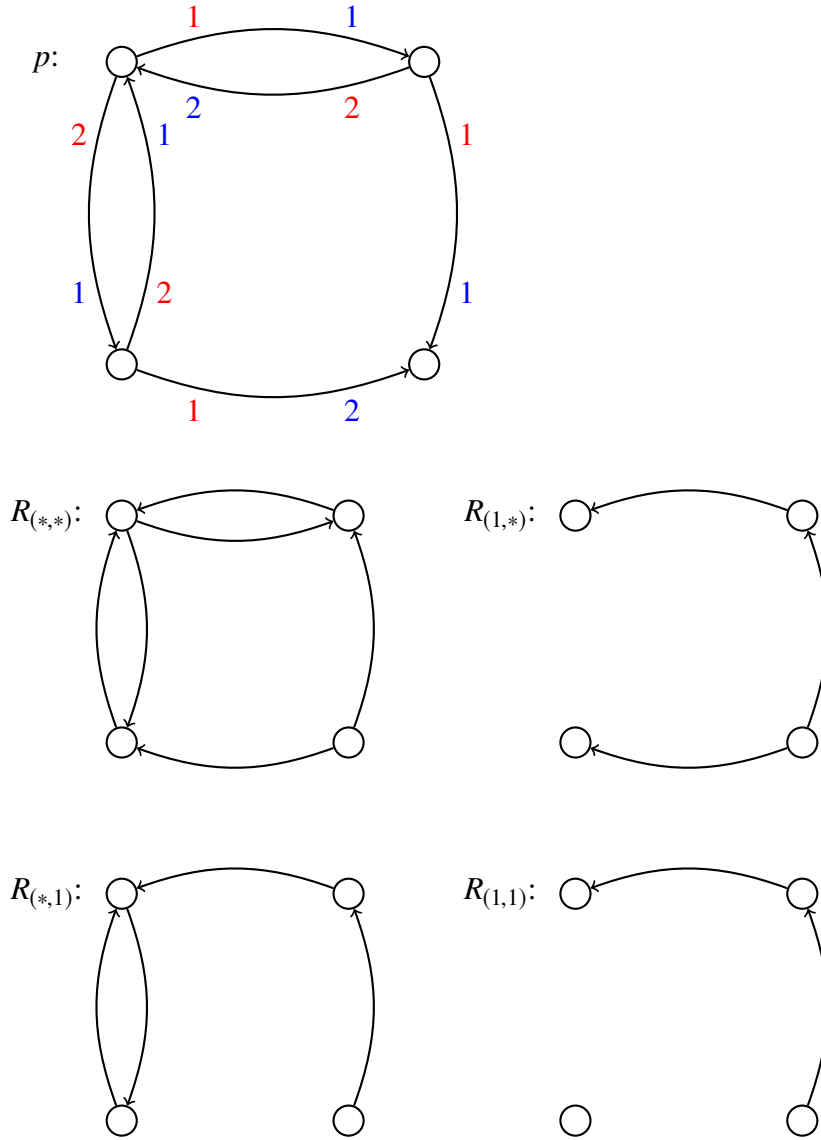
Merkitsemme $\mathfrak{M}_{a^*,b^*} = \{\mathcal{M}_{a^*,b^*}(G, p) \mid G \text{ on graafi ja } p \text{ sen porttinumerointi}\}$, kun $a^* \in \{a, *\}$ ja $b^* \in \{b, *\}$, sekä $\mathfrak{M}_{a,b}^c = \{\mathcal{M}_{a,b}(G, p) \mid G \text{ on graafi ja } p \text{ sen konsistentti porttinumerointi}\}$.

Ennen kuin voimme muotoilla vakioaikaisten luokkien yhteyden modaali-logiikoihin, joudumme määrittelemään vielä lisää käsitteitä.

Määritelmä 4.1. Vrt. [6, s. 15]. Olkoon $Y = Y_0 \uplus Y_1$, $a^* \in \{a, *\}$, $b^* \in \{b, *\}$ ja $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\}$, missä $\psi_{a,b} \in \text{GMML}(I_{a,b}, \Phi_{a,b})$ kaikilla $a, b \in \mathbb{Z}_+$. Sanomme, että Ψ määrittelee ratkaisun graafiongelmalle Π luokassa \mathfrak{M}_{a^*,b^*} , mikäli kaikilla luvuilla $a, b \in \mathbb{Z}_+$, kaikilla graafeilla $G = (V, E) \in \mathcal{F}(a, b)$ ja kaikilla graafin G porttinumeroinneilla p pätee, että on olemassa sellainen $S \in \Pi(G)$, että $S^{-1}[Y_1] = \|\psi_{a,b}\|_{\mathcal{M}_{a^*,b^*}(G,p)}$. Määrittelemme graafiongelman ratkaisun luokassa $\mathfrak{M}_{a,b}^c$ samalla tavalla.

Määritelmän ehdossa siis tarkoitetaan, että kaavan totuus mallin pisteessä on samastettavissa tilanteeseen, jossa sama solmu päättyy tilaan $y \in Y_1$ ajettaessa kaavaa vastaavaa algoritmia.

Huomautus. Kun $Y = Y_0 \uplus Y_1$, niin jokainen kaavajoukko $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\}$ määrää kanonisen graafiongelman Π^Ψ , jolle se määrittelee ratkaisun: Π^Ψ liittyy jokaiseen graafin $G = (V, E) \in \mathcal{F}(a, b)$ täsmälleen ne kuvaukset S , joilla $S^{-1}[Y_1] = \|\psi_{a,b}\|_{\mathcal{M}_{a^*,b^*}(G,p)}$ jollakin graafin G porttinumeroinnilla p .



Kuva 4.1: Ylimmässä kuvassa graafi ja sen porttinumerointi. Alemmissä kuvissa graafia vastaavan mallin relaatiot $R_{(*,*)}$, $R_{(1,*)}$, $R_{(*,1)}$ ja $R_{(1,1)}$.

Määritelmä 4.2. Vrt. [6, s. 15-16]. Olkoon \mathcal{L} jokin modaalilogiikoista ML, MML, GML ja GMML, olkoon C jokin graafiongelmien luokka, $a^* \in \{a, *\}$ ja $b^* \in \{b, *\}$. Sanomme, että \mathcal{L} karakterisoi luokan C luokassa \mathfrak{M}_{a^*, b^*} , jos seuraavat kaksi ehtoa pätevät:

- (i) Jos $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\}$ on joukko kaavoja, jolle pätee $\psi_{a,b} \in \mathcal{L}(\mathbf{I}_{a^*, b^*}, \Phi_{a,b})$ kaikilla $a, b \in \mathbb{Z}_+$, niin $\Pi^\Psi \in C$.
- (ii) Jos $\Pi \in C$ on graafiongelma, niin on olemassa joukko kaavoja $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\}$, missä $\psi_{a,b} \in \mathcal{L}(\mathbf{I}_{a^*, b^*}, \Phi_{a,b})$ kaikilla $a, b \in \mathbb{Z}_+$, joka määrittelee ratkaisun graafiongelmalle Π luokassa \mathfrak{M}_{a^*, b^*} .

Määrittelemme luokan C karakterisoinnin luokassa $\mathfrak{M}_{a,b}^c$ samalla tavalla.

Nyt voimme todistaa, että eri modaalilogiikat karakterisoivat vakioaikaiset graafiongelma-
luokat sopivissa luokissa \mathcal{M}_{a^*, b^*} , missä $a^* \in \{a, *\}$ ja $b^* \in \{b, *\}$. Käytämme lähtöporttien tie-
don sisältäviä malleja karakterisoimaan luokkia, joissa solmut voivat lähettää eri lähtöportteihin

eri viestejä. Käytämme tuloporttien tiedon sisältäviä malleja ja porrastettuja modaalilogiikoita karakterisoimaan luokkia, joissa solmut voivat laskea saamansa viestit. Multimodaalilogiikkaa käytämme erottamaan eritavoin numeroidut särmät toisistaan.

Lause 4.3. *Vrt. [6, s. 16].*

- (i) MML karakterisoi luokan $VV_c(1)$ luokassa $\mathfrak{M}_{a,b}^c$.
- (ii) MML karakterisoi luokan $VV(1)$ luokassa $\mathfrak{M}_{a,b}$.
- (iii) GMML karakterisoi luokan $MV(1)$ luokassa $\mathfrak{M}_{a,*}$.
- (iv) MML karakterisoi luokan $SV(1)$ luokassa $\mathfrak{M}_{a,*}$.
- (v) MML karakterisoi luokan $VB(1)$ luokassa $\mathfrak{M}_{*,b}$.
- (vi) GML karakterisoi luokan $MB(1)$ luokassa $\mathfrak{M}_{*,*}$.
- (vii) ML karakterisoi luokan $SB(1)$ luokassa $\mathfrak{M}_{*,*}$.

Todistus. Vrt. [6, s. 16-19]. Todistamme väitteen kohdan (iii), samaan tapaan kuin lähteessä todistetaan kohta (ii). Muut kohdat ovat myös todistettavissa samaan tapaan.

Olkoon $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\}$ joukko kaavoja, jolle pätee $\psi_{a,b} \in \text{GMML}(I_{a,*}, \Phi_{a,b})$ kaikilla $a, b \in \mathbb{Z}_+$. Määrittelemme kanonisen graafiongelman Π^Ψ lokaalin algoritmin $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{M}$ siten, että jokainen $\mathcal{A}_{a,b} \in \mathcal{A}$ simuloi kaavan $\psi_{a,b}$ totuuden määrittämistä mallissa $\mathcal{M}_{a,*}(G, p)$ kaikilla graafeilla $G \in \mathcal{F}(a, b)$ ja niiden porttinumeroinneilla p .

Olkoon $D_i = \{\eta \in \text{sf}(\psi_{a,b}) \mid \langle (i, *) \rangle_{\geq k} \eta \in \text{sf}(\psi_{a,b}) \text{ ja } k \in \mathbb{Z}_+\}$ kaikilla $i \in [a]$. Määrittelemme algoritmin $\mathcal{A}_{a,b}$ tilojen joukon Z , lopputilojen joukon $Y \subseteq Z$ ja viestien joukon M seuraavasti:

- $Z := \{f \mid f: \text{sf}(\psi_{a,b}) \rightarrow \{0, 1, U\}\},$
- $Y = Y_0 \uplus Y_1 := \{f \in Z \mid f(\psi_{a,b}) = 0\} \uplus \{f \in Z \mid f(\psi_{a,b}) = 1\},$
- $M := \bigcup_{i \in a} \{h \mid h: D_i \rightarrow \{0, 1, U\} \times \{i\}\} \cup \{m_0\}.$

Tavoitteenamme on luoda hajautettu algoritmi, missä solmun v tila ajanhetkellä t , $x_t(v)$, on kuvaus, joka kuvaa kaikki kaavan $\psi_{a,b}$ alikaavat, joiden modaalinen syvyys on korkeintaan t , arvoille 0 ja 1 sen mukaan, ovatko alikaavat solmussa epätosia (0) vai tosia (1). Alikaavat, joiden modaalinen syvyys on enemmän kuin t , kuvautuvat arvolle U , joka merkitsee totuusarvon määrittämättömyyttä (undefined). Osaa tiloista Z ei koskaan käytetä, kuten niitä kuvauksia, jotka kuvaavat alikaavat keskenään ristiriitaisille totuusarvoille. Tästä ei kuitenkaan koidu haittaa. Toisin sanoen tavoittelemme algoritmia, jossa solmun v tila ajanhetkellä t on $x_t(v) = f_{v,t} \in Z$, missä kuvaukselle $f_{v,t}$ pätee:

$$f_{v,t}(\eta) = \begin{cases} 0, & \text{kun } \text{md}(\eta) \leq t \text{ ja } \mathcal{M}_{a,*}(G, p), v \neq \eta \\ 1, & \text{kun } \text{md}(\eta) \leq t \text{ ja } \mathcal{M}_{a,*}(G, p), v \vDash \eta \\ U, & \text{kun } \text{md}(\eta) > t \end{cases}$$

missä $\eta \in \text{sf}(\psi_{a,b})$.

Kuvauksen $z_0: \{0, 1, \dots, a\} \times \{0, 1, \dots, b\} \rightarrow Z$, joka antaa solmujen alkutilat $x_0(v) = z_0(\text{deg}_+(v), \text{deg}_-(v))$, määrittelemme seuraavasti: Olkoot $\gamma, \theta \in \text{sf}(\psi_{a,b})$. Jokaisella $(i, j) \in$

$\{0, 1, \dots, a\} \times \{0, 1, \dots, b\}$ määrittelemme, että $z_0(i, j) = g$, missä $g \in Z$ on kuvaus, jonka määrittelemme rekursiivisesti:

$$\begin{aligned}
g(q_{i,j}) &= 1, \text{ jos } q_{i,j} \in \text{sf}(\psi_{a,b}). \\
g(q_{i',j'}) &= 0, \text{ jos } q_{i',j'} \in \text{sf}(\psi_{a,b}) \text{ ja } (i, j) \neq (i', j'). \\
g(\perp) &= 0, \text{ jos } \perp \in \text{sf}(\psi_{a,b}). \\
g(\neg\gamma) &= \begin{cases} 0, & \text{kun } g(\gamma) = 1 \\ 1, & \text{kun } g(\gamma) = 0 \\ U, & \text{kun } g(\gamma) = U \end{cases} \\
g(\gamma \vee \theta) &= \begin{cases} 0, & \text{kun } g(\gamma) = g(\theta) = 0 \\ 1, & \text{kun } 1 \in \{g(\gamma), g(\theta)\} \subseteq \{0, 1\} \\ U, & \text{kun } g(\gamma) = U \text{ tai } g(\theta) = U \end{cases} \\
g(\langle\alpha\rangle_{\geq k}\gamma) &= U \text{ kaikilla } \alpha \in I_{a,*}.
\end{aligned}$$

Siis jos solmun lähtöaste on i ja tuloaste j , niin $z_0(i, j) = g$ kuvaa totuusarvon 0 tai 1 kaikille alikaavoille, joiden modaalinen syvyys on 0. Ainoa propositiosymboli, joka saa arvokseen 1 on $q_{i,j}$, joka on siis solmussa tosi. Muut propositiosymbolit ovat vastaavasti solmussa epätosia. Kaikki kaavat, joiden modaalinen syvyys on enemmän kuin 0, saavat arvokseen U .

Seuraavaksi määrittelemme kuvauksen $\mu: Z \times [a] \rightarrow M$, joka määrää kussakin tilassa kuhunkin lähtöpörttiin lähetettävän viestin. Jos $f \in Z$ ja $i \in [a]$, niin $\mu(f, i) = h$, missä $h: D_i \rightarrow \{0, 1, U\} \times \{i\}$, $h(\eta) = (f(\eta), i)$, eli h on kuvauksen f rajoittuma joukkoon D_i , lisäksi kaavoissa esiintyvän lähtöpörtin indeksi i . Siis viesti $\mu(f, i) = h$ sisältää tiedon kaikkien kaavojen totuusarvosta, jotka johonkin modaaliopeattoreista yhdistettynä ovat kaavan $\psi_{a,b}$ alikaavoja. Viestit näin ollen välittävät solmujen kesken kaiken tiedon, jonka ne tarvitsevat muista solmuista kaavan $\psi_{a,b}$ totuuden määrittämiseen.

Viimeisenä algoritmin $\mathcal{A}_{a,b}$ muodostamiseen määrittelemme kuvauksen $\delta: Z \times M^b \rightarrow Z$, joka siirtää solmut tilasta toiseen. Olkoon $f \in Z$ ja $\vec{h} \in M^b$. Mikäli $f(\psi_{a,b}) \in \{0, 1\}$, niin $\delta(f, \vec{h}) = f$. Muussa tapauksessa määrittelemme, että $\delta(f, \vec{h}) = g$, missä g on kuvaus, jonka määrittelemme seuraavasti:

- $g(\eta) = f(\eta)$ kaikilla $\eta \in \text{sf}(\psi_{a,b})$, joille pätee $f(\eta) \neq U$.
- Olkoon $\eta \in \text{sf}(\psi_{a,b})$, jolle $f(\eta) = U$. Määrittelemme arvon $g(\eta)$ rekursiivisesti. Mikäli $\eta = \perp$ tai $\eta = q_{i,j}$ joillakin $i \in \{0, 1, \dots, a\}$ ja $j \in \{0, 1, \dots, b\}$, niin $g(\eta) = f(\eta)$. Muussa tapauksessa määrittelemme rekursiivisesti:

$$\begin{aligned}
g(\neg\gamma) &= \begin{cases} 0, & \text{kun } f(\gamma) = 1 \\ 1, & \text{kun } f(\gamma) = 0 \\ U, & \text{kun } f(\gamma) = U \end{cases} \\
g(\gamma \vee \theta) &= \begin{cases} 0, & \text{kun } f(\gamma) = f(\theta) = 0 \\ 1, & \text{kun } 1 \in \{f(\gamma), f(\theta)\} \subseteq \{0, 1\} \\ U, & \text{kun } f(\gamma) = U \text{ tai } f(\theta) = U \end{cases} \\
g(\langle\langle i, * \rangle\rangle_{\geq k}\gamma) &= \begin{cases} 0, & \text{kun } \sum_{h \in M} c_h > b - k, \text{ missä } (h, c_h) \in \text{ms}(\vec{h}) \\ & \text{ja joko } h(\gamma) = (0, i) \text{ tai } \gamma \notin \text{dom}(h) \text{ tai } h = m_0 \\ 1, & \text{kun } \sum_{h \in M} c_h \geq k, \text{ missä } (h, c_h) \in \text{ms}(\vec{h}) \text{ ja } h(\gamma) = (1, i) \\ U, & \text{muulloin} \end{cases}
\end{aligned}$$

Siis kuvaus, joka siirtää solmun tilasta toiseen, laskee joka askeleella yhtä modaalisyvyyttä suurempien kaavojen totuusarvon ja jättää jo aiemmissa vaiheissa laskettujen kaavojen totuusarvot ennalleen. Nyt hajautettu algoritmi $\mathcal{A}_{a,b} = (Z, M, z_0, \mu, \delta) \in \mathbf{M}$, sillä siirtymäkuvausten δ arvot eivät riipu saatujen viestien järjestyksestä, vaan ainoastaan niiden lukumääristä.

Seuraavaksi todistamme, että muodostamamme hajautetut algoritmit $\mathcal{A}_{a,b}$ todellakin vastaavat kaavan $\psi_{a,b}$ totuuden määrittämistä kaikilla $a, b \in \mathbb{Z}_+$. Olkoot siis $a, b \in \mathbb{Z}_+$ ja $G = (V, E) \in \mathcal{F}(a, b)$ graafi sekä olkoon p graafin G porttinumerointi. Osoitamme, että jos $v \in V$, $\eta \in \text{sf}(\psi_{a,b})$, $\text{md}(\eta) \leq t \leq \text{md}(\psi_{a,b})$ ja $x_t(v) = f \in Z$, niin $f(\eta) \in \{0, 1\}$ ja $f(\eta) = 1$, jos ja vain jos $\mathcal{M}_{a,*}(G, p), v \vDash \eta$.

Oletetaan, että $v \in V$ ja $x_t(v) = f \in Z$. Todistetaan väite induktiolla kaavan η rakenteen suhteen:

- Oletetaan, että $\eta = q_{i,j} \in \Phi_{a,b}$. Olkoot $\text{deg}_+(v) = i'$ ja $\text{deg}_-(v) = j'$. Jos $i = i'$ ja $j = j'$, niin $f(q_{i,j}) = (z_0(i, j))(q_{i,j}) = 1$. Jos taas $i \neq i'$ tai $j \neq j'$, niin $f(q_{i,j}) = (z_0(i', j'))(q_{i,j}) = 0$. Siis $f(q_{i,j}) \in \{0, 1\}$. Lisäksi $f(q_{i,j}) = (z_0(i', j'))(q_{i,j}) = 1$, jos ja vain jos $i = i'$ ja $j = j'$, jos ja vain jos $v \in \tau(q_{i,j})$, jos ja vain jos $\mathcal{M}_{a,*}(G, p), v \vDash q_{i,j}$.
- Jos $\eta = \perp$, niin $f(\perp) = (z_0(i', j'))(\perp) = 0 \in \{0, 1\}$ ja $\mathcal{M}_{a,*}(G, p), v \not\vDash \perp$.

Teemme induktio-oletuksen, että väite pätee kaavoille γ ja θ .

- Jos $\eta = \neg\gamma$, niin $\gamma \in \text{sf}(\eta) \subseteq \text{sf}(\psi_{a,b})$ ja $\text{md}(\gamma) = \text{md}(\eta) \leq t \leq \text{md}(\psi_{a,b})$. Induktio-oletuksen nojalla $f(\gamma) \in \{0, 1\}$. Jos $f(\gamma) = 0$, niin $f(\neg\gamma) = 1$ ja jos $f(\gamma) = 1$, niin $f(\neg\gamma) = 0$. Siis $f(\neg\gamma) \in \{0, 1\}$. Lisäksi $f(\neg\gamma) = 1$, jos ja vain jos $f(\gamma) = 0$. Tämä on induktio-oletuksen perusteella yhtäpitävää sen kanssa, että $\mathcal{M}_{a,*}(G, p), v \not\vDash \gamma$, minkä kanssa on yhtäpitävää, että $\mathcal{M}_{a,*}(G, p), v \vDash \neg\gamma$.
- Jos $\eta = \gamma \vee \theta$, niin $\gamma, \theta \in \text{sf}(\eta) \subseteq \text{sf}(\psi_{a,b})$ ja $\text{md}(\gamma), \text{md}(\theta) \leq \text{md}(\eta) \leq t \leq \text{md}(\psi_{a,b})$. Induktio-oletuksen nojalla $f(\gamma), f(\theta) \in \{0, 1\}$. Jos $f(\gamma) = f(\theta) = 0$, niin $f(\gamma \vee \theta) = 0$ ja jos $1 \in \{f(\gamma), f(\theta)\}$, niin $f(\gamma \vee \theta) = 1$. Lisäksi $f(\gamma \vee \theta) = 1$, jos ja vain jos $f(\gamma) = 1$ tai $f(\theta) = 1$. Tämä on induktio-oletuksen perusteella yhtäpitävää sen kanssa, että $\mathcal{M}_{a,*}(G, p), v \vDash \gamma$ tai $\mathcal{M}_{a,*}(G, p), v \vDash \theta$, mikä puolestaan on yhtäpitävää sen kanssa, että $\mathcal{M}_{a,*}(G, p), v \vDash \gamma \vee \theta$.
- Jos $\eta = \langle\langle(i, *)\rangle_{\geq k}\gamma\rangle$, niin $\gamma \in \text{sf}(\eta) \subseteq \text{sf}(\psi_{a,b})$ ja $\text{md}(\gamma) = \text{md}(\eta) - 1 < \text{md}(\eta) \leq t \leq \text{md}(\psi_{a,b})$. Jos $|R_{(i,*)}[v]| < k$, niin $f(\eta) = 0$ ja $\mathcal{M}_{a,*}(G, p), v \not\vDash \langle\langle(i, *)\rangle_{\geq k}\gamma\rangle$. Jos taas $|R_{(i,*)}[v]| \geq k$, niin induktio-oletuksen nojalla $g_u(\gamma) \in \{0, 1\}$ kaikilla $u \in R_{(i,*)}[v]$, missä $g_u = x_t(u)$. Jos $|\{u \in R_{(i,*)}[v] \mid g_u(\gamma) = 1\}| \geq k$, niin $f(\langle\langle(i, *)\rangle_{\geq k}\gamma\rangle) = 1$. Jos sen sijaan $|\{u \in R_{(i,*)}[v] \mid g_u(\gamma) = 1\}| = l < k$, niin $|\{u \in R_{(i,*)}[v] \mid g_u(\gamma) = 0\}| = |R_{(i,*)}[v]| - l$ ja $|\{w \in R_{(i',*)}[v] \mid i' \in a \setminus \{i\}\}| = \text{deg}_-(v) - |R_{(i,*)}[v]|$. Tällöin $|\{u \in R_{(i,*)}[v] \mid g_u(\gamma) = 0\}| + |\{w \in R_{(i',*)}[v] \mid i' \in a \setminus \{i\}\}| = (|R_{(i,*)}[v]| - l) + (\text{deg}_-(v) - |R_{(i,*)}[v]|) = \text{deg}_-(v) - l > \text{deg}_-(v) - k$, joten $f(\langle\langle(i, *)\rangle_{\geq k}\gamma\rangle) = 0$. Nyt pätee $f(\langle\langle(i, *)\rangle_{\geq k}\gamma\rangle) = 1$, jos ja vain jos $|\{u \in R_{(i,*)}[v] \mid g_u(\gamma) = 1\}| \geq k$. Induktio-oletuksen nojalla tämä on yhtäpitävää sen kanssa, että $|\{u \in R_{(i,*)}[v] \mid \mathcal{M}_{a,*}(G, p), u \vDash \gamma\}| \geq k$, mikä on edelleen yhtäpitävää sen kanssa, että $\mathcal{M}_{a,*}(G, p), v \vDash \langle\langle(i, *)\rangle_{\geq k}\gamma\rangle$.

Induktiodistuksen perusteella jos $\text{md}(\psi_{a,b}) = T$ ja $x_T(v) = f$, niin $f(\psi_{a,b}) \in \{0, 1\}$ ja $f(\psi_{a,b}) = 1$, jos ja vain jos $\mathcal{M}_{a,*}(G, p), v \vDash \psi_{a,b}$. Koska tämä pätee kaikilla solmuilla $v \in V$, niin algoritmi $\mathcal{A}_{a,b}$ pysähtyy ajanhetkellä T ja $S_{\mathcal{A}_{a,b}, (G, p)}^{-1}[Y_1] = \|\psi_{a,b}\|^{\mathcal{M}_{a,*}(G, p)}$. Koska tämä pätee kaikilla graafeilla $G \in \mathcal{F}(a, b)$, porttinumeroinneilla p ja hajautetuilla algoritmeilla $\mathcal{A}_{a,b} \in \mathcal{A}$,

ja jokainen T riippuu vain vastaavan kaavan syvyydestä, niin \mathcal{A} on graafiongelman Π^Ψ lokaali algoritmi ja $\Pi^\Psi \in \text{MV}(1)$.

Nyt olemme todistaneet väitteen ensimmäiseen suuntaan (määritelmän 4.2 kohta (i)). Seuraavaksi todistamme väitteen toisen suunnan (määritelmän 4.2 kohta (ii)). Tätä varten tarvitsemme kaksi uutta käsitettä. Jos Ψ on joukko porrastetun modaalilogiikan kaavoja, niin niiden *Boolean kombinaatiot* ovat kaavoja, jotka on rakennettu kaavoista $\psi \in \Psi$ käyttämällä näiden kaavojen lisäksi ainoastaan konnektiiveja \neg ja \vee . Porrastetun modaalilogiikan kaavojen ψ ja θ sanotaan olevan *loogisesti ekvivalentit*, jos kaikilla malleilla \mathcal{M} ja pisteillä $w \in W$ pätee, että $\mathcal{M}, w \models \psi$, jos ja vain jos $\mathcal{M}, w \models \theta$. Näiden käsitteiden avulla voimme ryhtyä todistamaan väitteen toista suuntaa.

Oletetaan, että $\Pi \in \text{MV}(1)$. Siis on olemassa sellainen graafiongelman Π lokaali algoritmi $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{M}$, että jos $G = (V, E) \in \mathcal{F}(a, b)$ ja p on graafin G porttinumerointi, niin $\mathcal{S}_{\mathcal{A}_{a,b},(G,p)} \in \Pi(G)$. Tavoitteenamme on luoda sellainen joukko kaavoja $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \text{GMML}(\mathbb{I}_{a,*}, \Phi_{a,b})$, että jos $v \in V$, niin $\mathcal{M}_{a,*}(G, p), v \models \psi_{a,b}$, jos ja vain jos $\mathcal{S}_{\mathcal{A}_{a,b},(G,p)}(v) \in Y_1$, missä Y_1 on hajautetun algoritmin $\mathcal{A}_{a,b}$ hyväksyvien lopputilojen joukko. Toisin sanoen jokaiselle hajautetulle algoritmillemme rakennetaan sellainen kaava, että algoritmin ajon tulostetta graafissa vastaa kaavan totuusarvot graafia vastaavan mallin pisteissä.

Olkoon $\mathcal{A}_{a,b} = (Z, M, z_0, \mu, \delta)$ ja $Y_0 \uplus Y_1 \subseteq Z$. Olkoon T algoritmin $\mathcal{A}_{a,b}$ ajoon kuuluva aika. Rakennamme algoritmin $\mathcal{A}_{a,b}$ ajoa simuloivan kaavan $\psi_{a,b} \in \text{GMML}(\mathbb{I}_{a,*}, \Phi_{a,b})$ alikaavoista $\gamma_{z,t}$, $\theta_{m,i,t}$ ja $\sigma_{m,i,t}$.

Kaavan $\gamma_{z,t}$ totuus graafia vastaavan mallin pisteessä merkitsee, että pistettä vastaava solmu on ajanhetkellä t tilassa z . Kaavan $\theta_{m,i,t}$ totuus pisteessä merkitsee, että vastaava solmu lähettää ajanhetkellä t viestin m lähtöporttiinsa i . Kaavan $\sigma_{m,i,t}$ totuus vastaavasti merkitsee, että kyseinen solmu saa ajanhetkellä t viestin m solmulta, joka lähetti kyseisen viestin lähtöporttiinsa i . Rakennamme nämä kaavat rekursiivisesti:

- $\gamma_{z,0}$ on propositiosymbolien $q_{i,j} \in \Phi_{a,b}$ tietty Boolean kombinaatio, kun $z \in Z$,
- $\theta_{m,i,t+1}$ on kaavojen $\gamma_{z,t}$ (missä $z \in Z$) tietty Boolean kombinaatio, kun $m \in M$, $i \in [a]$ ja $t \in \{0, 1, \dots, T\}$,
- $\sigma_{m,i,t+1}$ on kaavojen $\langle\langle i, * \rangle\rangle_{\geq k} \theta_{m,i,t+1}$ (missä $k \geq 0$) tietty Boolean kombinaatio, kun $m \in M$ ja $t \in \{0, 1, \dots, T\}$,
- $\gamma_{z,t+1}$ on kaavojen $\gamma_{x,t}$ ($x \in Z$) ja kaavojen $\sigma_{m,i,t+1}$ (missä $m \in M$ ja $i \in [a]$) tietty Boolean kombinaatio, kun $z \in Z$ ja $t < T$.

Mitkä tahansa Boolean kombinaatiot eivät todellakaan käy. Kaavojen valtavan koon vuoksi niitä on kuitenkin hankala esittää formaalisti, joten muotoilemme niitä tässä sanallisesti mahdollisimman tarkasti. Olemme käyttäneet tähän mennessä ainoastaan lauselogiikan konnektiiveja \neg ja \vee . Käytämme Boolean kombinaatioissa konjunktioita, \wedge , joka saadaan lyhenteenä konnektiiveista \neg ja \vee seuraavasti: $\phi \wedge \phi' := \neg(\neg\phi \vee \neg\phi')$. Sen totuus merkitsee, että molemmat kaavoista ϕ ja ϕ' ovat tosia. Muodostaessamme rekursiossa esiintyvät Boolean kombinaatiot, käymme läpi yksitellen jokaisen niihin käytettävän kaavan ja otamme siitä joko kaavan itse tai sen negaation, riippuen siitä, onko kaava tarkasteltavassa solmussa tosi vai epätosi. Yhdistämme nämä toisiinsa konjunktioilla. Kyseiset Boolean kombinaatiot siis kertovat täydellisen tiedon solmuissa tosista (tarkastelun kannalta oleellisista) kaavoista ajanhetken rajaaman modaalisen syvyyden mukaan. Esimerkiksi jos $\Phi_{a,b} = \{q_{0,0}, q_{0,1}, q_{1,0}, q_{1,1}\}$, $z \in Z$ ja $\gamma_{z,0} = \neg q_{0,0} \wedge q_{0,1} \wedge \neg q_{1,0} \wedge \neg q_{1,1}$, niin kaavan $\gamma_{z,0}$ totuus solmussa kertoisi, että siinä on tosi täsmälleen propositiosymboli $q_{0,1}$, eli sen lähtöaste olisi 0 ja tuloaste 1.

Etsimämme kaava $\psi_{a,b} \in \text{GMML}(\mathbb{I}_{a,*}, \Phi_{a,b})$ on nyt siis disjunktio kaavoista $\gamma_{y,T}$, missä $y \in Y_1$. Kaava voidaan kuitenkin muodostaa vain siinä tapauksessa, että alikaavoja on äärellinen määrä. Mikäli joko Z tai M on ääretön, saa näistä rekursion avulla tuotettua äärettömän määrän alikaavoja. Voimme kuitenkin todistaa, että on vain äärellinen määrä alikaavoja, jotka eivät ole keskenään loogisesti ekvivalentteja. Todistamme tämän väitteen joukoille $\Gamma_t = \{\gamma_{z,t} \mid z \in Z\}$, $\Theta_t = \{\theta_{m,i,t} \mid m \in M, i \in [a]\}$ ja $\Sigma_t = \{\sigma_{m,i,t} \mid m \in M, i \in [a]\}$ induktiolla ajanhetken t suhteen:

- Olkoon $t = 0$. Nyt kaikki kaavat $\gamma_{z,0} \in \Gamma_0$ ovat propositiosymbolien $q_{i,j} \in \Phi_{a,b}$ Boolean kombinaatioita. Kukin $\gamma_{z,0}$ pystyy ilmaisemaan ainoastaan propositiosymbolien totuusarvoja tarkasteltavassa pisteessä. Koska $\Phi_{a,b}$ on äärellinen ja mahdollisia totuusarvoja ovat vain 0 ja 1, niin kaavat $\gamma_{z,0}$ ovat samaistettavissa kuvauksiin $\Phi_{a,b} \rightarrow \{0, 1\}$, joita on äärellinen määrä. Siis joukossa Γ_0 on äärellinen määrä kaavoja, jotka eivät ole keskenään loogisesti ekvivalentteja.

- Olkoon $t = 1$. Nyt kaikki kaavat $\theta_{m,i,1} \in \Theta_1$ ovat kaavojen $\gamma_{z,0}$ Boolean kombinaatioita. Koska kaavoja $\gamma_{z,0}$, jotka eivät ole keskenään ekvivalentteja, on äärellinen määrä, niin niiden keskenään ei-ekvivalentteja Boolean kombinaatioitakin on vain äärellinen määrä. Siis joukossa Θ_1 on vain äärellinen määrä kaavoja, jotka eivät ole keskenään ekvivalentteja.

Kaikki kaavat $\sigma_{m,i,1} \in \Sigma_1$ ovat nyt kaavojen $\langle (i, *) \rangle_{\geq k} \theta_{m,i,1}$ Boolean kombinaatioita, missä $i \in [a]$ ja $k \geq 0$. Keskenään ei-ekvivalentteja kaavoja $\theta_{m,i,1}$ on äärellinen määrä. Arvoja $k \geq 0$ on periaatteessa ääretön määrä, mutta kun $k \geq b$, niin arvon k suuruus ei enää vaikuta kaavan totuusarvoon mallissa $\mathcal{M}_{a,*}(G, p)$, kun $G \in \mathcal{F}(a, b)$. Siispä joukossa Σ_1 on vain äärellinen määrä keskenään ei-ekvivalentteja kaavoja.

Nyt kaavat $\gamma_{z,1} \in \Gamma_1$ ovat kaavojen $\gamma_{x,0}$ ja $\sigma_{m,i,1}$ Boolean kombinaatioita. Koska keskenään ei-ekvivalentteja kaavoja $\gamma_{x,0}$ ja $\sigma_{m,i,1}$ on vain äärellinen määrä, niin samoin on vain äärellinen määrä niiden keskenään ei-ekvivalentteja Boolean kombinaatioita. Siis joukossa Γ_1 on vain äärellinen määrä kaavoja, jotka eivät ole keskenään ekvivalentteja.

- Oletetaan, että väite pätee ajanhetkellä t . Samoin kuin vaiheessa $t = 1$, joukkojen Θ_{t+1} , Σ_{t+1} ja Γ_{t+1} kaavat ovat Boolean kombinaatioita äärellisestä määrästä keskenään ei-ekvivalentteja kaavoja, joten näitä keskenään ei-ekvivalentteja Boolean kombinaatioitakin on vain äärellinen määrä. Siis joukoissa Θ_{t+1} , Σ_{t+1} ja Γ_{t+1} on vain äärellinen määrä keskenään ei-ekvivalentteja kaavoja.

Nyt jos hajautetun algoritmin $\mathcal{A}_{a,b}$ syötteenä on (G, p) , missä p on graafin $G = (V, E) \in \mathcal{F}(a, b)$ porttinumerointi ja $v \in V$, niin $S_{\mathcal{A}_{a,b},(G,p)}(v) \in Y_1$, jos ja vain jos $v \in \|\gamma_{y,T}\|^{\mathcal{M}_{a,*}(G,p)}$ jollakin $y \in Y_1$. Koska joukko Y_1 on äärellinen, voimme merkitä $Y_1 = \{y_1, y_2, \dots, y_n\}$, missä $n \in \mathbb{Z}_+$. Nyt voimme määritellä $\psi_{a,b} := \gamma_{y_1,T} \vee \gamma_{y_2,T} \vee \dots \vee \gamma_{y_n,T}$. Siis $S_{\mathcal{A}_{a,b},(G,p)}^{-1}[Y_1] = \|\psi_{a,b}\|^{\mathcal{M}_{a,*}(G,p)}$ kaikilla graafeilla $G \in \mathcal{F}(a, b)$ ja niiden porttinumeroinneilla p . Näin ollen kaavajoukko $\Psi = \{\psi_{a,b} \mid a, b \in \mathbb{Z}_+\}$ määrittelee ratkaisun graafiongelmalle Π luokassa $\mathfrak{M}_{a,*}$.

Osoitamme lopuksi induktiolla, että $\text{md}(\gamma_{z,t}) = t$ kaikilla $\gamma_{z,t} \in \Gamma_t$.

- Kaavat $\gamma_{z,0}$ ovat propositiosymbolien Boolean kombinaatioita, joten $\text{md}(\gamma_{z,0}) = 0$ kaikilla $\gamma_{z,0} \in \Gamma_0$.
- Oletetaan, että $\text{md}(\gamma_{x,t}) = t$ kaikilla $\gamma_{x,t} \in \Gamma_t$. Nyt kaavat $\theta_{m,i,t+1}$ ovat kaavojen $\gamma_{x,t}$ Boolean kombinaatioita, joten $\text{md}(\theta_{m,i,t+1}) = t$ kaikilla $\theta_{m,i,t+1} \in \Theta_{t+1}$. Koska kaavat $\sigma_{m,i,t+1}$ ovat kaavojen $\langle (i, *) \rangle_{\geq k} \theta_{m,i,t+1}$ Boolean kombinaatioita, niin $\text{md}(\sigma_{m,i,t+1}) = t + 1$ kaikilla $\sigma_{m,i,t+1} \in \Sigma_{t+1}$. Siis koska kaavat $\gamma_{z,t+1}$ ovat kaavojen $\gamma_{x,t}$ ja $\sigma_{m,i,t+1}$ Boolean kombinaatioita, niin $\text{md}(\gamma_{z,t+1}) = t + 1$ kaikilla $\gamma_{z,t+1} \in \Gamma_{t+1}$.

Tästä seuraa erityisesti, että $\text{md}(\psi_{a,b}) = \text{md}(\gamma_{y_1,T} \vee \gamma_{y_2,T} \vee \dots \vee \gamma_{y_n,T}) = T$ eli algoritmin $\mathcal{A}_{a,b}$ ajoon kuuluva aika. \square

Olemme nyt karakterisoineet jokaisen graafiongelmaluokan. Vaikka karakterisointi on samanlainen kuin lähteessä [6], tulemme kuitenkin seuraavassa luvussa osoittaneeksi, että raja suunnattuihin graafeihin heikentää luokkien välisiä yhteyksiä. On syytä perehtyä suunnattujen ja suuntaamattomien graafien välisiin eroihin hajautettujen algoritmien näkökulmasta.

Yamashita ja Kameda [9, s.71-74, Luku 3] osoittivat, että yksinkertaisissa graafissa kaikki se tieto, jonka solmut algoritmin ajon aikana keräävät, on pelkästään puutteellista tietoa syötegraafin topologiasta. Saamiensa viestien perusteella solmut saavat tiedon ns. lokaaleista näköpiireistään, eli äärellisistä puista, joiden juurena on solmu itse ja joissa voidaan tehdä syötegraafia vastaavia siirtymiä solmusta toiseen. Siis jos syötegraafissa on polku, jonka pituus on t lähtien solmusta v , niin ajanhetkellä t solmun v muodostamassa puussa on olemassa sellainen lehtisolmu, että suora polku juuresta lehteen vastaa kyseistä polkua syötegraafissa. Jos yksinkertaisen graafin pisimmän suoran polun pituus on k , niin ajanhetkellä k solmujen muodostamat lokaalit näköpiirit sisältävät ainakin yhden kopion jokaisesta graafin solmusta ja näin ollen bisimilaarisuutta vaille täydellisen tiedon koko graafin topologiasta.

Suunnattujen graafien tapauksessa solmujen näköpiirit sen sijaan ovat suunnattuja puita (Esimerkki 3.13). Toisin kuin yhtenäisten yksinkertaisten graafien tapauksessa, yhtenäisissä suunnatuissa graafeissa on mahdollista, että kahden solmun välillä ei ole minkäänlaista polkua, jolloin kyseiset solmut eivät muodosta kopioita toisistaan näköpiireissään. Näin ollen solmujen keräämä tieto voi suunnatuissa graafeissa olla huomattavasti puutteellisempaa kuin vastaavissa yksinkertaisissa graafeissa. Lisäksi, mikäli suunnatussa graafissa ei ole silmukoita ja sen pisimmän polun pituus on k , niin graafin solmujen muodostamat näköpiirit eivät enää laajene ajanhetken k jälkeen, eivätkä solmut saa enää mitään uutta informaatiota kyseisen ajanhetken jälkeen.

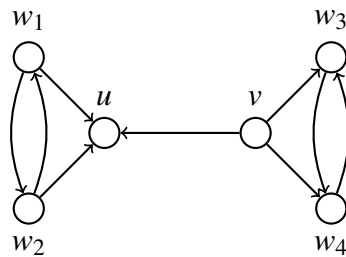
Voimme tutkia eroavaisuuksia vastaavasti myös modaalilogiikan näkökulmasta. Suunnatut graafit asettavat rajoituksen niitä vastaaville Kripke-malleille: kahden pisteen välillä voi olla enintään yksi relaatio kumpaankin suuntaan. Yksinkertaisten graafien tapauksessa tämä rajoitus yhä pätee, mutta vastaavien mallien pitää olla myös tietystä mielessä symmetrisiä: jos kahden pisteen välillä kulkee relaatio yhteen suuntaan, pitää niiden välillä kulkea myös relaatio toiseen suuntaan, mutta relaatioiden ei tarvitse olla samoja. Tämänlainen symmetrisyys kuitenkin takaa, että jos pisteessä v on tosi kaava ϕ , niin pisteessä w on tosi kaava $\langle \alpha_1 \rangle \dots \langle \alpha_n \rangle \phi$, missä $\alpha_1, \dots, \alpha_n$ vastaavat käänteisessä järjestyksessä kulkua relaatioita pitkin pisteestä w pisteeseen v . Näin pisteet saavat tietoa kaikista muista mallin pisteistä. Kaava $\diamond \rightarrow \diamond \rightarrow \perp$ on esimerkki modaalilogiikan kaavasta, joka voi toteutua suunnattua graafia vastaavassa mallissa, mutta ei yksinkertaista graafia vastaavassa mallissa.

Esimerkki 4.4. Tarkastelemme kuvan 4.2 graafia. Graafissa kiinnostuksen kohteena ovat solmut u ja v . Solmu v kykenee lähettämään viestejä, mutta ei vastaanottamaan niitä. Solmu u sen sijaan päinvastoin kykenee ainoastaan vastaanottamaan viestejä. Solmun v kaltaisia solmuja kutsumme *lähteiksi* (*source*) ja solmun u kaltaisia solmuja *nieluiksi* (*sink*). Tällaiset solmut ovat suunnatuille graafeille ominaisia, eikä vastaavaa ilmiötä esiinny yksinkertaisissa graafeissa.

Luonteensa vuoksi solmu u saa tietoa ainoastaan graafin solmuista v , w_1 ja w_2 , joten solmut w_3 ja w_4 eivät voi mitenkään vaikuttaa solmun u lokaaliin tulosteeseen. Solmu u ei myöskään kykene vaikuttamaan minkään muun solmun lokaaliin tulosteeseen. Solmu v sen sijaan ei saa tietoa mistään graafin muista solmuista. Erityisesti, ajettiin graafissa mikä tahansa hajautettu algoritmi, joka pysähtyy, niin solmu v on lopputilassa viimeistään ajanhetkellä 1. Seurauksena

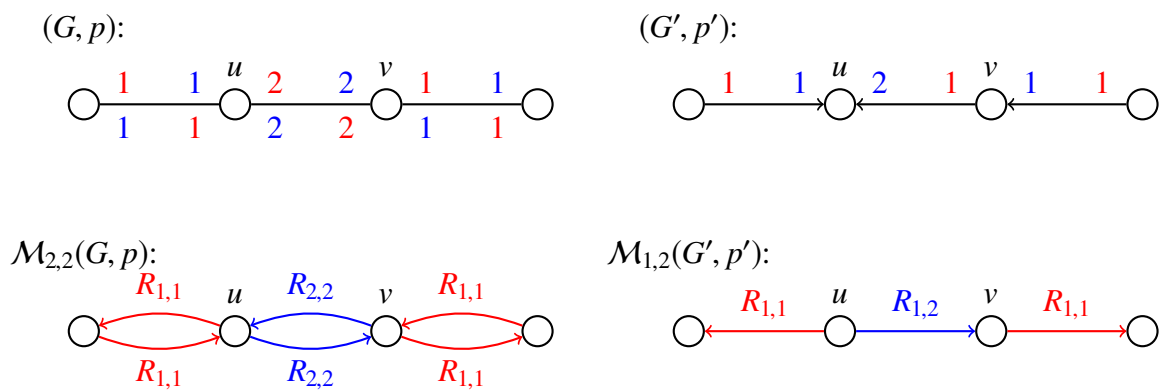
solmut w_1 ja w_2 eivät saa tietoa solmuista w_3 ja w_4 ja päinvastoin. Siis nämä solmuparit suorittavat laskennan toisistaan riippumattomina.

Yksinkertaisten graafien tapauksessakin on mahdollista luoda sellaisia algoritmeja, että tietyt solmut eivät tietyissä graafeissa pääse lähettämään tai vastaanottamaan tietoa, esimerkiksi määrittelemällä, että tietyllä asteella solmu lähettää vain viestiä m_0 . Mikä tekee tästä ilmiöstä erityisen suunnatuille graafeille on, että tämä rajoitus pätee kaikille hajautetuille algoritmeille, jotka graafeissa voi ajaa. Tämän vuoksi mikään graafiongelma, joka vaatisi solmun tietävän jotain kaikista graafin solmuista, ei ole ratkaistavissa ilman erityisiä rajoituksia hyväksyttävälle sytegraafeille (esimerkiksi oletus vahvasta yhtenäisyydestä). Vaihtoehtoisesti graafien solmuille voidaan antaa ylimääräistä tietoa, kuten Esimerkissä 3.13.



Kuva 4.2: Suunnattu graafi, jossa mikään solmuista ei saa tietoa kaikista muista solmuista.

Suunnatuilla graafeilla on siis yksinkertaisista graafeista poikkeavia ominaisuuksia, joita voidaan tutkia hajautetuilla algoritmeilla. Vaihtoehtoinen lähestymistapa olisi tutkia yksinkertaisten graafien ominaisuuksia suunnattujen graafien avulla. Kaikki yksinkertaiset graafit voidaan muuntaa suunnatuiksi graafeiksi valitsemalla jokaiselle särmälle erikseen suunta. Näitä vastaavia suunnattuja graafeja kutsutaan alkuperäisen graafin orientaatioiksi. Erityisesti olisi mielenkiintoista, jos orientaatioiden avulla voitaisiin ratkaista graafiongelmia, jotka eivät ratkea yksinkertaisilla graafeilla. Seuraavassa esimerkissä esitämme tällaisen graafiongelman.



Kuva 4.3: Vasemmalla metsä G ja sen konsistentti porttinumerointi p sekä metsää G vastaava Kripke-malli. Oikealla eräs metsän G orientaatio ja sitä vastaava Kripke-malli.

Esimerkki 4.5. Tässä tutkielmassa tarkastelemme yksinkertaisia graafeja suunnattujen graafien erikoistapauksena. Voimme samaistaa yksinkertaiset graafit sellaisiin suunnattuihin graafeihin $G = (V, E)$, joille pätee, että $(v, v) \notin E$ ja jos $(u, v) \in E$, niin $(v, u) \in E$ kaikilla $u, v \in V$. Näin tulkittuina määrittelemämme algoritmien ja graafiongelmien luokat yhtenevät Hellan et al. [6] esittämiin luokkiin. Suunnattu graafi $D = (V', E')$ on yksinkertaisen graafin G orientaatio,

mikäli $V = V'$ ja kaikilla $(u, v) \in E$ pätee joko $(v, u) \in E'$ ja $(u, v) \notin E'$ tai $(u, v) \in E'$ ja $(v, u) \notin E'$ [4, s. 171]. Yksinkertainen graafi on *metsä*, mikäli siinä ei ole silmukkaa (eli suuntaamatonta suoraa polkua solmusta itseensä, jonka pituus on vähintään 3) [4, s. 26]. Graafin G *solmujen k -väritys* on kuvaus $V \rightarrow [k]$ [4, s. 117]. Määrittämme siis jokaiselle solmulle yhden värin, joita merkitsemme $\{1, \dots, k\}$.

Tarkastelemamme graafiongelma on sellaisen mielivaltaisen k -värityksen löytäminen metsälle, missä mikään solmu ei ole samanvärinen minkään naapurinsa kanssa. Määrittelemme graafiongelman seuraavasti: Jos $G = (V, E)$ on metsä (vastaavasti metsän orientaatio) ja $S: V \rightarrow \mathbb{Z}_+$, niin $S \in \Pi(G)$, jos ja vain jos seuraava ehto pätee kaikilla $v \in V$:

- $S(v) \neq S(u)$ aina, kun $(v, u) \in E$.

Tällainen väritys toki löytyy jokaisesta graafista, kun esimerkiksi kaikki solmut väritetään eri väreillä. Hajautettujen algoritmien tapauksessa ongelma on kuitenkin ylivoimainen yksinkertaisille graafeille jopa luokassa VV_c . Riittää tarkastella kuvan 4.3 graafin G solmuja u ja v . Näemme, että sitä vastaavassa Kripke-mallissa $\mathcal{M}_{2,2}(G, p)$ pisteet u ja v ovat bisimilaarisia multimodaalilogiikan suhteen. Siis ajettiinpa graafissa mikä algoritmi tahansa, solmut u ja v päätyvät aina samaan lopputilaan, eli ne saavat saman värin, vaikka ne ovat toistensa naapureita.

Metsien orientaatioille ongelma on kuitenkin suorastaan ominainen, sillä se ratkeaa jopa luokassa SB. Määrittelemme perheen hajautettuja algoritmeja $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{S} \cap \mathbf{B}$, joka ratkaisee graafiongelman Π . Olkoon $\mathcal{A}_{a,b} = (Z, M, z_0, \mu, \delta)$. Määrittelemme joukot Z ja M sekä kuvaukset z_0 ja μ seuraavasti:

- $Z = \mathbb{N}$, missä tilat \mathbb{Z}_+ ovat lopputiloja eli värejä ja 0 on väliaikainen tila.
- $M = Z \cup \{m_0\}$.
- $z_0: \{0, \dots, a\} \times \{0, \dots, b\} \rightarrow Z$, missä $z_0(i, 0) = 1$ ja $z_0(i, j) = 0$ kaikilla $i \in \{0, \dots, a\}$ ja $j \neq 0$. Siis lähteet alustetaan lopullisella värillä 1 ja loput solmuista jätetään ilman väriä.
- $\mu: Z \times [a]$, $\mu(z, i) = z$ kaikilla $i \in [a]$. Siis solmut kommunikoivat tilojensa välityksellä.

Lopuksi määrittelemme kuvauksen $\delta: Z \times M^b \rightarrow Z$:

$$\delta(z, (m_1, \dots, m_b)) = \begin{cases} \min(\mathbb{Z}_+ \setminus \{m_1, \dots, m_b\}), & \text{kun } z = 0 \text{ ja } m_j \neq 0 \text{ kaikilla } j \in [b] \\ 0, & \text{kun } z = 0 \text{ ja } m_j = 0 \text{ jollakin } j \in [b] \\ z, & \text{kun } z \in \mathbb{Z}_+ \end{cases}$$

Algoritmi toimii seuraavasti: Aluksi jokainen syötegraafin lähde väritetään värillä 1. Tällaisia solmuja on aina olemassa, sillä syötegraafit ovat orientoituja metsiä. Mikäli suunnatun graafin jokaisen solmun tuloaste olisi positiivinen, niin pätsi yksi kahdesta tapauksesta: joko graafi olisi ääretön tai siinä olisi silmukka. Äärettömiä graafeja emme tarkastele, kun taas silmukka orientoituisessa graafissa merkitsisi vastaavan yksinkertaisen graafin sisältävän silmukan. Tällöin yksinkertainen graafi ei voisi olla metsä.

Laskennan seuraavassa vaiheessa solmut välittävät samansisältöisen viestin omasta tilastaan jokaiseen lähtöpörttiin. Mikäli solmun v kaikki vastaanottamat viestit kertovat viestin lähtäneen solmun olevan väritetty, niin solmu v värityy pienimmällä mahdollisella värillä, joka poikkeaa sen saamista viesteistä. Näin väritys etenee orientaatioiden lähteistä sen nieluihin. Metsien orientaatioissa kaikki lähteestä alkavat polut päätyvät ennen pitkää nieluun, samalla

argumentilla kuin osoitimme lähteiden olemassaolon: muussa tapauksessa orientaatio olisi ääretön tai siinä olisi silmukka. Lisäksi jokainen orientaation solmu kuuluu johonkin tällaiseen polkuun, joten koko orientaatio värityy. Jokainen solmuista väritetään eri tavalla kuin sitä välittömästi edeltäneet solmut ja sen seuraajat myös poikkeavat valitusta väristä. Siis saamamme k -väritys on halutunlainen.

Huomautus. Edeltävä esimerkki osoittaa, että särmien suuntaaminen rikkoo vierekkäisten solmujen symmetrisyyden metsissä. Rajoitus metsiin on tarpeellinen, sillä tietyt silmukan sisältävät graafit voidaan orientoida siten, että vierekkäiset solmut ovat symmetrisiä: esimerkiksi kolmen solmun silmukan kellonsuuntaisessa orientaatioissa kaikki solmut ovat keskenään symmetrisiä, eikä siinä ole yhtään lähdettä tai nielua. Algoritmit, joissa laskenta etenee lähteistä nieluihin, eivät voi koskaan päättyä vakioajassa, mutta ne ovat suunnatuille graafeille ominainen tapa suorittaa laskentoja graafeissa, joissa ei ole suunnattuja silmukoita.

5 Graafiongelmien luokkien väliset identiteetit ja eroavaisuudet

Luvussa 3.5 esitimme seuraavat graafiongelmien luokkien väliset osajoukkorelaatiot:

- (i) $SB \subsetneq MB = VB \subsetneq MV = VV \subsetneq VV_c$,
- (ii) $SB \subsetneq SV \subsetneq MV = VV \subsetneq VV_c$,
- (iii) $SB(1) \subsetneq MB(1) = VB(1) \subsetneq MV(1) = VV(1) \subsetneq VV_c(1)$,
- (iv) $SB(1) \subsetneq SV(1) \subsetneq MV(1) = VV(1) \subsetneq VV_c(1)$.

Tässä luvussa todistamme ensimmäiseksi relaatioissa esiintyvät luokkien identiteetit $MB = VB$ ja $MV = VV$. Luvun ja tutkielman lopuksi osoitamme relaatioissa esiintyvät luokkien eroavaisuudet $SB \neq MB$, $SV \neq MV$, $SB \neq SV$, $VB \neq MV$ ja $VV \neq VV_c$ sekä luokkien lineaarisuuden rikkovat tulokset $SV \not\subseteq VB$ ja $VB \not\subseteq SV$ käyttämällä hyväksi bisimulaatiota luvun 4 karakterisoinnin mukaisesti. Samaan aikaan tulemme luvun aikana todistaneeksi kaikki samat tulokset vakioaikaisille luokille.

5.1 Luokkien väliset identiteetit

Tässä luvussa todistamme, että kaikki graafiongelmat, jotka ratkeavat algoritmeilla, joissa solmut vastaanottavat viestinsä vektoreina, ratkeavat myös, jos viestit vastaanotettaisiin multijoukkoina (olettamatta, että porttinumeroinnit ovat konsistentteja). Esitämme yksityiskohtaisen todistuksen tulokselle $MV = VV$ ja perustelemme tuloksen $MB = VB$ vain lyhyesti todistusten samanlaisuuden vuoksi.

Lause 5.1. *Vrt. [6, s. 21]. Olkoon Π graafiongelma ja $T: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$. Olkoon $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{V}$ perhe hajautettuja algoritmeja, joka ratkaisee graafiongelman Π ajassa T . Tällöin on olemassa perhe hajautettuja algoritmeja $\mathcal{A}' = \{\mathcal{A}'_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{M}$, joka ratkaisee graafiongelman Π ajassa T .*

Todistus. Vrt. [6, s. 21-22]. Olkoon $G = (V, E) \in \mathcal{F}(a, b)$ graafi ja p sen porttinumerointi. Tarkastelemme hajautetun algoritmin $\mathcal{A}_{a,b} = (Z, M, z_0, \mu, \delta)$ ajoa (G, p) :ssä. Määrittelemme jokaisella $t \in \mathbb{N}$ kuvauksen $\beta_t: P_-(G) \rightarrow M^t$ seuraavasti:

$$\beta_t(j, u) = (h_1(j, u), h_2(j, u), \dots, h_t(j, u)),$$

missä $u \in V$, $j \in [\deg_-(u)]$ ja $h_l(j, u)$ on se viesti, minkä solmu u vastaanotti tuloportista (j, u) ajanhetkellä l . Siis kuva $\beta_t(j, u)$ on vektori kaikista viesteistä, jotka solmu u on vastaanottanut portista (j, u) ajanhetkeen t mennessä.

Seuraavaksi määrittelemme näille vektoreille sanakirjajärjestyksen $<$ seuraavasti: Olkoon $<_M$ jokin viestijoukon M lineaarijärjestys. Olkoon sitten $t \in \mathbb{N}$, $u \in V$ ja $j, j' \in [\deg_-(u)]$. Määrittelemme, että $\beta_t(j, u) < \beta_t(j', u)$, jos ja vain jos on olemassa $l \in [t]$, jolle $h_l(j, u) <_M h_l(j', u)$ ja $h_k(j, u) = h_k(j', u)$ kaikilla $k \in [l - 1]$. Toisin sanoen vertaamme vektoreiden $\beta_t(j, u)$ ja $\beta_t(j', u)$ viestejä vanhimmasta uusimpaan ja järjestämme vektorit ensimmäisen eroavan viestiparin mukaan joukon M lineaarijärjestyksen perusteella. Siis vektorien järjestys on samanlainen kuin sanojen järjestys sanakirjassa.

Lopuksi määrittelemme, että porttinumerointi p on yhteensopiva viestihistorian kanssa ajanhetkellä t , mikäli $\beta_t(j, u) \leq \beta_t(j', u)$ kaikilla $u \in V$ ja $j, j' \in [\deg_-(u)]$, missä $j < j'$. Huomaamme, että mikäli p on yhteensopiva viestihistorian kanssa ajanhetkellä t , on se myös yhteensopiva viestihistorian kanssa ajanhetkellä l kaikilla $l \in [t]$.

Nyt olemme valmiita muodostamaan algoritmin $\mathcal{A}'_{a,b}$, joka simuloi algoritmin $\mathcal{A}_{a,b}$ ajoa (G, p) :ssä jollain porttinumeroinnilla p . Olkoon p_0 graafin G porttinumerointi, jota tarkastelemme algoritmin $\mathcal{A}'_{a,b}$ yhteydessä. Olkoon P_0 kaikkien niiden porttinumerointien p joukko, joille pätee seuraava ehto:

kaikilla $v \in V$ ja $i \in [\deg_+(v)]$ on olemassa sellaiset $u \in V$ ja $j, j' \in [\deg_-(u)]$, että

$$p(v, i) = (j, u) \text{ ja } p_0(v, i) = (j', u).$$

Kokoamme siis joukkoon P_0 kaikki ne porttinumeroinnit, jotka numeroivat lähtöportit samalla tavoin vastaanottavien solmujen suhteen porttinumeroinnin p_0 kanssa. Algoritmit, jotka kuuluvat luokkaan \mathbf{M} , eivät siis näe niiden välillä mitään eroa. Havaitsemme joukkoa P_0 koskien seuraavat neljä asiaa, kun ajamme algoritmia $\mathcal{A}'_{a,b}$:

- (a) P_0 on epätyhjä ja jokainen $p \in P_0$ on yhteensopiva viestihistorian kanssa ajanhetkellä 0.
- (b) Graafin G solmujen tilat ajanhetkellä 0 eivät riipu porttinumeroinnin $p \in P_0$ valinnasta.
- (c) Solmun v porttiin (v, i) lähettämä viesti ajanhetkellä 1 ei riipu porttinumeroinnin $p \in P_0$ valinnasta.
- (d) On olemassa ainakin yksi $p \in P_0$, joka on yhteensopiva viestihistorian kanssa ajanhetkellä 1.

Havainto (a) seuraa joukon P_0 määritelmästä, havainto (b) seuraa kuvauksen z_0 määritelmästä ja havainto (c) seuraa havainnosta (b) ja kuvauksen μ määritelmästä. Havainto (d) taas seuraa, kun huomaamme, että solmun eri tuloporteista vastaanottamat viestit ajanhetkellä 1 voidaan aina järjestää lineaarijärjestyksen $<_M$ perusteella.

Määrittelemme vielä rekursiivisesti kaikilla $t \in \mathbb{N}$, että $P_t \subseteq P_{t-1}$ on niiden porttinumerointien joukko, jotka ovat yhteensopivia viestihistorian kanssa ajanhetkellä t . Teemme seuraavat neljä havaintoa joukosta P_t induktiivisesti:

- (i) P_t on epätyhjä ja jokainen $p \in P_t$ on yhteensopiva viestihistorian kanssa ajanhetkellä t .
- (ii) Solmun $u \in V$ ajanhetkellä t vastaanottama viestien vektori $\vec{h}_t(u)$ ei riipu porttinumeroinnin $p \in P_t$ valinnasta. Graafin G solmujen tilat ajanhetkellä t eivät riipu porttinumeroinnin $p \in P_t$ valinnasta.
- (iii) Solmun v porttiin (v, i) lähettämä viesti ajanhetkellä $t + 1$ ei riipu porttinumeroinnin $p \in P_t$ valinnasta.
- (iv) On olemassa ainakin yksi $p \in P_t$, joka on yhteensopiva viestihistorian kanssa ajanhetkellä $t + 1$.

Havainto (i) perustuu havaintoon (iv) joukolle P_{t-1} ja havainto (ii) perustuu havaintoon (iii) joukolle P_{t-1} . Havainto (iii) seuraa havainnosta (ii) ja kuvauksen μ määritelmästä. Lopuksi havainto (iv) seuraa siitä, että solmun u vastaanottamat viestihistoriat $\beta_t(j, u)$ voidaan aina järjestää. Kyseistä järjestystä vastaava porttinumerointi p on yhteensopiva viestihistorian kanssa ajanhetkellä l ja edelleen $p \in P_l$, kaikilla $l \in [t]$.

Olkoon $T = T(a, b, |V|)$. Koska \mathcal{A} ratkaisee graafiongelman Π ajassa T , niin algoritmi $\mathcal{A}_{a,b}$ pysähtyy ajassa $T(G, p)$:ssä kaikilla $p \in P_T$. Porttinumeroinnit $p \in P_T$ olemme muodostaneet siten, että lähtöporttien numerot on kopioitu porttinumeroinnista p_0 ja tuloportteja numeroidessa olemme hyväksyneet vain ne porttinumeroinnit, jotka ovat yhteensopivia viestihistorian kanssa ajanhetkellä T . Nyt ajettaessa algoritmia $\mathcal{A}'_{a,b}$, mikäli kukin solmu liittää lähettämäänsä viestiin kaikki aikaisemmat samasta lähtöportista lähetetyt viestit aikajärjestyksessä, niin kukin solmu pystyy järjestämään vastaanottamansa viestihistoriat siten, että jos $\beta_t(j, u) < \beta_t(j', u)$, niin $j < j'$ ja jos $\beta_t(j, u) = \beta_t(j', u)$, niin $j = j'$. Siis jos solmu u vastaanottaa (G, p_0) :ssa ajanhetkellä t viestien multijoukon

$$\text{ms}(\beta_t(1, u), \beta_t(2, u), \dots, \beta_t(b, u)),$$

niin se pystyy järjestämään viestit $\beta_t(j, u)$ tavalla, joka vastaa viestien vektoria, jonka solmu u vastaanottaisi (G, p) :ssä, $p \in P_t$, kun siinä ajettaisiin algoritmi $\mathcal{A}_{a,b}$.

Määrittelemme siis, että $\mathcal{A}'_{a,b} = (Z', M', z'_0, \mu', \delta')$, joka johdetaan algoritmista $\mathcal{A}_{a,b}$ seuraavasti:

- Viestien joukko M' on viestien $m \in M$ vektorit $\{m' \in M^t \mid t \in \mathbb{N}\}$.
- Tilojen joukko Z' pitää kirjaa kuhunkin lähtöporttiin lähetetyistä viesteistä, eli $Z' = Z \times M'^a$.
- Alkutilat ovat samat kuin kuvauksen z_0 määräämät, lisänä tyhjät viestivektorit kullekin lähtöportille.
- Solmun u porttiin i tilassa $(z, \vec{m}_1, \dots, \vec{m}_a)$ lähetetty viesti on vektori \vec{m}_i , jonka loppuun lisätään joukon M viesti $\mu(z, i)$.
- Jos solmu u saa tilassa $(z, \vec{m}_1, \dots, \vec{m}_a)$ viestit $(\vec{h}_1, \dots, \vec{h}_b)$, se siirtyy tilaan $(z', \vec{m}'_1, \dots, \vec{m}'_a)$, missä $z' = \delta(z, (n_1, \dots, n_b))$, missä kukin n_j on vastaavan vektorin \vec{h}_j viimeinen alkio ja $\vec{m}'_i \in M'$ on sama kuin viesti, joka saadaan lisäämällä viestin \vec{m}_i loppuun $\mu(z, i)$.

Yksinkertaisemmin, algoritmi $\mathcal{A}'_{a,b}$ pitää tilojen kautta kirjaa kunkin solmun kuhunkin lähtöporttiin lähettämistä viesteistä. Lähetetyt viestit ja siirtymät tilasta toiseen tapahtuvat analogisesti algoritmin $\mathcal{A}_{a,b}$ kanssa. Jos algoritmista $\mathcal{A}_{a,b}$ solmu v lähettää porttiin i viestin m , niin algoritmi $\mathcal{A}'_{a,b}$ lähettää porttiin i vektorin algoritmista $\mathcal{A}_{a,b}$ jo aiemmin lähettämistään viesteistä, jonka loppuun lisätään viesti m . Siirryttäessä tilasta toiseen, kukin kirjaa pitävä viestivektori päivitetään viimeisellä kyseiseen porttiin lähetetyllä viestillä.

Nyt algoritmin $\mathcal{A}'_{a,b}$ ajo (G, p_0) :ssa simuloi algoritmin $\mathcal{A}_{a,b}$ ajoa (G, p) :ssä jollakin sopivalla $p \in P_T$. Algoritmin $\mathcal{A}_{a,b}$ tuloste $S_{\mathcal{A}_{a,b}, (G, p)}$ ei riipu valitusta porttinumeroinnista ja koska $S_{\mathcal{A}_{a,b}, (G, p)} \in \Pi(G)$, niin myös $S_{\mathcal{A}'_{a,b}, (G, p_0)} \in \Pi(G)$. Lisäksi koska algoritmi $\mathcal{A}_{a,b}$ pysähtyy ajassa T , niin myös algoritmi $\mathcal{A}'_{a,b}$ pysähtyy ajassa T . Koska tällainen algoritmi saadaan aikaan jokaisella luokalla $\mathcal{F}(a, b)$, niin meillä on perhe hajautettuja algoritmeja $\mathcal{A}' = \{\mathcal{A}'_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{M}$, joka ratkaisee graafiongelman Π ajassa T . \square

Lause 5.2. Vrt. [6, s. 22]. Olkoon Π graafiongelma ja $T: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$. Olkoon $\mathcal{A} = \{\mathcal{A}_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{B}$ perhe hajautettuja algoritmeja, joka ratkaisee graafiongelman Π ajassa T . Tällöin on olemassa perhe hajautettuja algoritmeja $\mathcal{A}' = \{\mathcal{A}'_{a,b} \mid a, b \in \mathbb{Z}_+\} \subseteq \mathbf{M} \cap \mathbf{B}$, joka ratkaisee graafiongelman Π ajassa T .

Todistus. Vrt. [6, s. 22]. Todistus on yksinkertaisempi versio lauseen 5.1 todistuksesta. Sen sijaan, että kukin solmu pitäisi monen vektorin avulla kirjata kaikista kustakin lähtöportista lähetetyistä viesteistä, riittää siihen tässä tapauksessa yksi vektori, sillä kustakin lähtöportista lähetetään samat viestit. Muuten todistus etenee samaan tapaan kuin lauseessa 5.1. \square

Seuraus 5.3. Vrt. [6, s. 22]. Tiedetään, että

- $MB = VB$ ja $MB(1) = VB(1)$,
- $MV = VV$ ja $MV(1) = VV(1)$.

Todistus. Vrt. [6, s. 22]. Triviaalisti pätee, että $MB \subseteq VB$ ja $MV \subseteq VV$. Lauseista 5.1 ja 5.2 seuraa, että $VB \subseteq MB$ ja $VV \subseteq MV$. Yhdessä näistä seuraa, että $MB = VB$ ja $MV = VV$.

Vakioaikaisten luokkien tapauksessa lauseista 5.1 ja 5.2 seuraa, että $VB(1) \subseteq MB(1)$ ja $VV(1) \subseteq MV(1)$, sillä todistuksissa muodostetut hajautetut algoritmit pysähtyvät samassa ajassa kuin alkuperäiset algoritmit. Erityisesti vakioaikaisten algoritmien tapauksessa ne pysähtyvät vakioajassa. Yhdessä triviaalien osajoukkorelaatioiden kanssa tästä seuraa, että $MB(1) = VB(1)$ ja $MV(1) = VV(1)$. \square

5.2 Luokkien väliset eroavaisuudet

Kun yhdistämme tietämämme graafiongelmaluokkien väliset triviaalit osajoukkorelaatiot ja seurauksen 5.3, niin seuraavat relaatiot pätevät luokkien välillä:

- (i) $SB \subseteq MB = VB \subseteq MV = VV \subseteq VV_c$,
- (ii) $SB \subseteq SV \subseteq MV = VV \subseteq VV_c$,
- (iii) $SB(1) \subseteq MB(1) = VB(1) \subseteq MV(1) = VV(1) \subseteq VV_c(1)$,
- (iv) $SB(1) \subseteq SV(1) \subseteq MV(1) = VV(1) \subseteq VV_c(1)$.

Jäljellä on ainoastaan osoittaa, että ketjuissa esiintyvien osajoukkorelaatioiden tapauksessa on kyse aidoista osajoukoista, sekä lineaarisuuden rikkovat tulokset:

- (1) $VB \not\subseteq SV$ ja $VB(1) \not\subseteq SV(1)$,
- (2) $SV \not\subseteq VB$ ja $SV(1) \not\subseteq VB(1)$.

Näiden tulosten todistamiseksi käytämme hyväksi bisimulaatioita ja luvun 4 tuloksia.

Ensimmäiseksi todistamme tutkielman tärkeimmän tuloksen, eli ensimmäisen lineaarisuuden rikkovan tuloksen $VB \not\subseteq SV$. Samalla tulemme todistaneeksi, että on olemassa graafiongelmia, jotka ratkeavat algoritmeilla, joissa viestit vastaanotetaan multijoukkoina, mutta eivät ratkea, mikäli viestit vastaanotetaan joukkoina. Toisin sanoen $SB \neq MB$ ja $SV \neq MV$. Tulokset $VB \not\subseteq SV$ ja $SV \neq MV$ poikkeavat Hellan et al. [6] artikkelista, missä osoitettiin, että $VB \subseteq SV$ ja $SV = MV$. Nämä tulokset eivät siis yleisty suunnatuille graafeille. Tulos $SB \neq MB$ esitettiin artikkelissa omana lauseenaan, mutta saamme sen tässä tapauksessa suoraviivaisena seurauksena.

Lause 5.4. On olemassa graafiongelma Π , jolle $\Pi \in VB(1)$, mutta $\Pi \notin SV$.

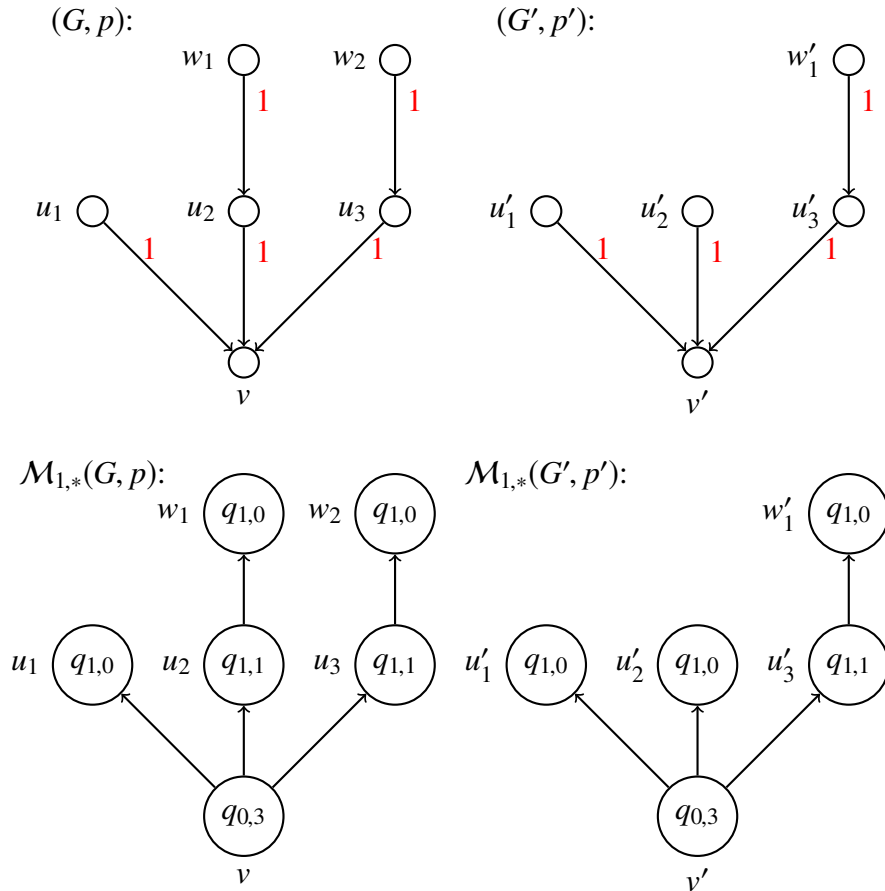
Todistus. Määrittelemme seuraavanlaisen graafiongelman Π : Jos $G = (V, E)$ on suunnattu graafi ja $S: V \rightarrow \{0, 1\}$, niin määrittelemme, että $S \in \Pi(G)$, jos ja vain jos seuraava ehto pätee kaikilla $v \in V$:

- $S(v) = 1$, jos ja vain jos on olemassa sellaiset solmut $u, w \in V$, että $(u, v), (w, v) \in E$ ja $\deg_-(u)$ sekä $\deg_-(w)$ ovat parillisia.

On helppoa osoittaa, että $\Pi \in \text{VB}(1)$. Ensimmäisessä laskennan vaiheessa jokainen solmu lähettää jokaista särmää pitkin samansisältöisen viestin, tiedon oman tuloasteensa parillisuudesta. Tämän jälkeen jokainen solmu laskee saamansa viestit, jotka kertovat tuloasteen olevan parillinen. Mikäli näitä viestejä on vähintään kaksi, siirtyy solmu tilaan 1. Jos niitä on vain yksi tai ei yhtään, solmu siirtyy tilaan 0. Laskentaan kuluva aika on vakio, sillä laskenta päättyy jo ajanhetkellä 1.

Näemme, että $\Pi \notin \text{SV}$ tarkastelemalla kuvan 5.1 graafeja G ja G' sekä niiden porttinumeroiteja p ja p' . Huomaamme, että $\deg_+(G) = \deg_+(G') = 1$ ja $\deg_-(G) = \deg_-(G') = 3$, joten $G, G' \in \mathcal{F}(1, 3)$. Graafin G solmulla v on olemassa vain yksi solmu u_1 , jolle $(u_1, v) \in E$ ja $\deg_-(u_1) = 0$ on parillinen. Graafin G' solmulla v' sen sijaan on kaksi solmua u'_1 ja u'_2 , joille $(u'_1, v'), (u'_2, v') \in E$ ja $\deg_-(u'_1) = \deg_-(u'_2) = 0$ ovat parillisia. Siis graafiongelman ratkaisevan algoritmin $\mathcal{A} \in \mathbf{S}$ tulisi hylätä solmu v ja hyväksyä solmu v' .

Huomaamme kuitenkin, että graafeja vastaavat mallit $\mathcal{M}_{1,*}(G, p)$ ja $\mathcal{M}_{1,*}(G', p')$ ovat keskenään bisimilaariset multimodaalilogiikan suhteen. Kyseinen bisimulaatio on relaatio $Z = \{(v, v'), (u_1, u'_1), (u_1, u'_2), (u_2, u'_3), (u_3, u'_3), (w_1, w'_1), (w_2, w'_1)\}$. Erityisesti pisteet v ja v' ovat keskenään bisimilaariset, joten niissä ovat tosia kaikki samat multimodaalilogiikan kaavat. Näin ollen algoritmia \mathcal{A} ajettaessa solmut v ja v' päätyisivät samaan lopputilaan. Siis $\Pi \notin \text{SV}$. \square



Kuva 5.1: Kuvassa yllä kaksi graafia, joiden alapuolella niitä vastaavat Kripke-mallit.

Huomautus. Tässä todistuksessa annetut mallit erottuvat luokassa $VB(1)$, sillä kyseinen luokka karakterisoidaan luokassa $\mathfrak{M}_{*,b}$, missä solmuihin saapuvat särmät jakautuvat eri relaatioiksi vastaavassa mallissa.

Seuraus 5.5. *Vrt. [6, s. 24]. Tiedetään, että*

- $VB \not\subseteq SV$ ja $VB(1) \not\subseteq SV(1)$,
- $SB \neq MB$ ja $SB(1) \neq MB(1)$,
- $SV \neq MV$ ja $SV(1) \neq MV(1)$.

Todistus. Vrt. [6, s. 24]. Ensinnäkin lauseen 5.4 graafiongelmalle Π pätee, että $\Pi \in VB(1) \subseteq VB$ ja $\Pi \notin SV \supseteq SV(1)$. Näin ollen $VB \not\subseteq SV$ ja $VB(1) \not\subseteq SV(1)$.

Toiseksi lauseen 5.2 perusteella $MB(1) = VB(1)$, joten $\Pi \in MB(1) \subseteq MB$. Toisaalta koska $SB \subseteq SV$, niin $\Pi \notin SB \supseteq SB(1)$. Tästä seuraa, että $SB \neq MB$ ja $SB(1) \neq MB(1)$.

Lopuksi, koska $VB(1) \subseteq VV(1) = MV(1)$, niin $\Pi \in MV(1) \subseteq MV$. Siis $SV \neq MV$ ja $SV(1) \neq MV(1)$. \square

Seuraavaksi todistamme toisen lineaarisuuden rikkovan tuloksen $SV \not\subseteq VB$. Samalla tulemme todistaneeksi, että on olemassa graafiongelmia, jotka ratkeavat algoritmeilla, joissa solmut kykenevät lähettämään eri viestejä eri solmuille, mutta eivät ratkea, mikäli solmujen täytyy lähettää kaikille solmuille sama viesti. Toisin sanoen $SB \neq SV$ ja $VB \neq MV$. Kaikki nämä tulokset ovat yhteneviä lähteen [6] kanssa, mutta on syytä todeta, että tuloksia $SB \neq SV$ ja $VB \neq MV$ ei mainita eksplisiittisesti kyseisessä artikkelissa, vaan ne olivat implisiittisiä seurauksia aiemmin mainituista vahvemmista tuloksista, jotka eivät yleisty suunnatuille graafeille.

Lause 5.6. *Vrt. [6, s. 23]. On olemassa graafiongelma Π , jolle $\Pi \in SV(1)$, mutta $\Pi \notin VB$.*

Todistus. Ks. [6, s. 23]. Rajoitumme tässä esimerkissä tarkastelemaan yhtenäisiä graafeja yksinkertaisuuden vuoksi. Sanomme, että yhtenäinen suunnattu graafi $G = (V, E)$ on (*suunnattu*) *k-tähtigraafi*, mikäli $V = \{c, v_1, \dots, v_k\}$ jollakin $k \in \{2, 3, \dots\}$ ja $E = \{(c, v_i) \mid i \in \{1, 2, \dots\}\}$. Solmua c kutsutaan tähden *keskipisteeksi* ja solmuja v_i kutsutaan sen *lehtisolmuiksi*. Todistukseksi tarkastelemme graafiongelmaa, jossa valitsemme yhden tähtigraafin lehtisolmuista.

Ensimmäiseksi osoitamme, että algoritmien $\mathcal{A} \in \mathbf{S} \cap \mathbf{B}$ on mahdollista selvittää, onko syötegraafi tähti. Määrittelemme graafiongelman Π_1 seuraavasti: Mikäli $G = (V, E)$ on yhtenäinen suunnattu graafi ja $S_1 : V \rightarrow \{C, L, F\}$, niin $S_1 \in \Pi_1(G)$, jos ja vain jos seuraavat ehdot pätevät kaikilla $v \in V$:

- $S_1(v) = C$, jos ja vain jos $\deg_-(v) = 0$.
- $S_1(v) = L$, jos ja vain jos $\deg_+(v) = 0$, $\deg_-(v) = 1$ ja solmulle $u \in V$, jolle $(u, v) \in E$, pätee $\deg_-(u) = 0$.
- $S_1(v) = F$, muulloin.

Lopputila C siis merkitsee tähden keskipisteen (center) ja lopputila L sen lehtisolmut (leaf). Lopputila F merkitsee solmut, jotka eivät voi olla osa tähtigraafia. Syötegraafi on tunnistettu tähtigraafiksi, mikäli $S_1 \in \Pi_1(G)$ ja $S_1(v) \neq F$ kaikilla $v \in V$.

Nyt näemme, että $\Pi_1 \in SB(1)$ tarkastelemalla seuraavanlaista algoritmia: Ensimmäisessä laskennan vaiheessa jokainen solmu lähettää jokaisesta lähtöportista samansisältöisen viestin omasta tuloasteestaan. Tämän jälkeen jokainen solmu tarkastaa oman lähtö- ja tuloasteensa sekä saamansa viestit. Mikäli sen tuloaste on 0, se siirtyy tilaan C . Jos taas sen tuloaste on 1 ja

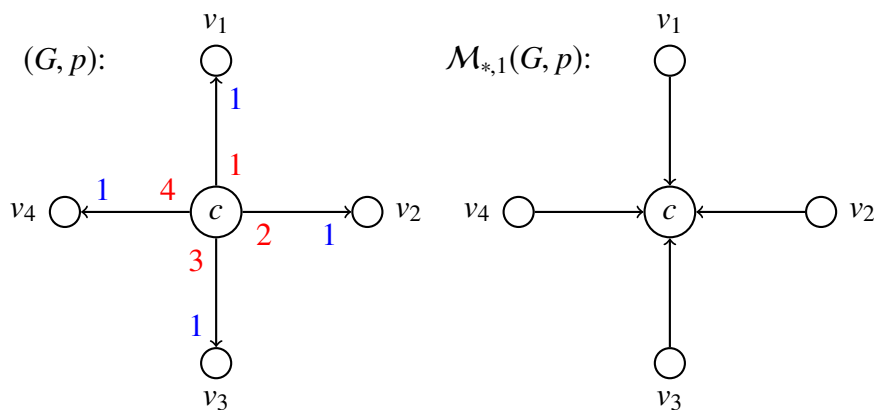
lähtöaste 0, sekä ainoa vastaanotettu viesti kertoo, että viestin lähettäneen solmun tuloaste on 0, siirtyy solmu tilaan L . Muussa tapauksessa solmu siirtyy tilaan F . Laskentaan kuuluva aika on vakio, sillä laskenta päättyy jälleen ajanhetkellä 1.

Nyt pystymme selvittämään kaikissa graafiongelmaluokissa, onko syötegraafi tähtigraafi. Tämä algoritmi voidaan suorittaa synkronisesti rinnakkain minkä tahansa muun algoritmin kanssa. Määrittelemme seuraavaksi yhdistetyn graafiongelman Π_2 seuraavasti: Jos $G = (V, E)$ on yhtenäinen suunnattu graafi ja $S_2: V \rightarrow \{0, 1\}$, niin $S_2 \in \Pi_2(G)$, mikäli toinen seuraavista ehtoista pätee:

- G on k -tähtigraafi, $S_2(c) = 0$ ja on olemassa sellainen $j \in [k]$, että $S_2(v_j) = 1$ ja $S_2(v_i) = 0$ kaikilla $i \neq j$.
- G ei ole tähtigraafi.

Näemme, että $\Pi_2 \in \text{SV}(1)$ seuraavasti: Ensimmäisessä laskennan vaiheessa jokainen solmu lähettää jokaisesta lähtöportista kyseisen lähtöportin numeron. Mikäli solmu saa viestin 1, se siirtyy tilaan 1. Jos se saa minkä tahansa muun viestin, se siirtyy tilaan 0. Tämä laskenta tapahtuu vakioajassa, sillä laskenta päättyy ajanhetkellä 1. Suorittamalla tämän algoritmin yhdessä aiemmin määrittelemämme algoritmin kanssa, saamme ajanhetkellä 1 tiedon siitä, onko syötegraafi tähti, ja mikäli se on, saamme samalla ajanhetkellä myös valituksi yhden sen lehtisolmuista.

Nähdäksemme, että $\Pi_2 \notin \text{VB}$, riittää tarkastella kuvan 5.2 graafia G ja sen porttinumerointia p . Luokalle $\text{SV}(1)$ määritelty algoritmi hyväksyisi vastaavan Kripke-mallin $\mathcal{M}_{4,*}(G, p)$ solmun v_1 ja hylkäisi kaikki muut, sillä ainoastaan solmu v_1 voi saada viestin, joka on lähetetty lähtöportista, jonka numero on 1. Kuitenkin tarkastellessamme graafia vastaavaa Kripke-mallia $\mathcal{M}_{*,1}(G, p)$, huomaamme kaikkien graafin lehtisolmujen olevan keskenään bisimilaarisia multimodaalilogiikan suhteen. Siis ajaessamme luokassa VB mitä algoritmia tahansa, graafin kaikki lehtisolmut päätyisivät samaan tilaan, joten algoritmi ei pystyisi erottamaan niitä toisistaan eikä valitsemaan niistä yhtäkään. Näin ollen $\Pi_2 \notin \text{VB}$. \square



Kuva 5.2: Kuvassa vasemmalla 4-tähtigraafi ja oikealla sitä vastaava Kripke-malli.

Huomautus. Lauseen 5.6 todistuksessa graafien oletettiin olevan yhtenäisiä. Tämä rajoitus ei ole välttämätön. Mikäli emme rajoittaisi syötegraafeja yhtenäisiin graafeihin, niin kuvaillut algoritmit yhä määrittäisivät, ovatko graafin kaikki komponentit tähtiä. Mikäli ne olisivat, valitsisivat algoritmit niistä jokaisesta yhden lehtisolmun. Laskenta tapahtuisi tällöinkin vakioajassa, sillä eri komponentit eivät voi vaikuttaa toisiinsa laskennan aikana. Muokkaamalla tutkittavan graafiongelman määritelmää saisimme näinkin aikaan graafiongelman Π , jolle $\Pi \in \text{SV}(1)$ ja $\Pi \notin \text{VB}$.

Seuraus 5.7. Vrt. [6, s. 23]. Tiedetään, että

- $SV \not\subseteq VB$ ja $SV(1) \not\subseteq VB(1)$,
- $SB \neq SV$ ja $SB(1) \neq SV(1)$,
- $VB \neq MV$ ja $VB(1) \neq MV(1)$.

Todistus. Vrt. [6, s. 23]. Ensimmäiseksi lauseen 5.6 graafiongelmalle Π pätee, että $\Pi \in SV(1) \subseteq SV$ ja $\Pi \notin VB \supseteq VB(1)$. Siis $SV \not\subseteq VB$ ja $SV(1) \not\subseteq VB(1)$.

Toiseksi, koska $SB \subseteq VB$, niin $\Pi \notin SB \supseteq SB(1)$. Näin ollen $SB \neq SV$ ja $SB(1) \neq SV(1)$.

Lopuksi, koska $SV(1) \subseteq MV(1)$, niin $\Pi \in MV(1) \subseteq MV$. Tästä seuraa, että $VB \neq MV$ ja $VB(1) \neq MV(1)$. \square

Lopuksi toteamme, että konsistenteilla porttinumeroilla voidaan ratkaista graafiongelmiä, jotka eivät ratkea, mikäli porttinumeroinnit eivät ole konsistentteja.

Lause 5.8. Vrt. [6, s. 24]. On olemassa graafiongelma Π , jolle $\Pi \in VV_c(1)$, mutta $\Pi \notin VV$.

Todistus. Ks. [6, s. 24-25]. Todistus on vastaava kuin lähteessä. Todistuksessa esitetty yksiner-tainen graafi riittää tulkita vastaavana suunnattuna ja symmetrisenä graafina. \square

Seuraus 5.9. Vrt. [6, s. 25]. Tiedetään, että $VV \neq VV_c$ ja $VV(1) \neq VV_c(1)$.

Todistus. Vrt. [6, s. 25]. Lauseen 5.8 graafiongelmalle Π pätee, että $\Pi \in VV_c(1) \subseteq VV_c$ ja $\Pi \notin VV \supseteq VV(1)$. Tästä seuraa, että $VV \neq VV_c$ ja $VV(1) \neq VV_c(1)$. \square

Olemme nyt todistaneet kaikki graafiongelmaluokkien väliset osajoukkorelaatiot. Näin ollen olemme osoittaneet, että kuvan 3.4 mukainen järjestys pätee luokkien välillä, sekä vastaavasti vakioaikaisten luokkien välillä. Esitämme lopuksi muutaman tästä johtuvan seurauksen.

Seuraus 5.10. Vrt. [6, s. 25].

- (a) Logiikan MML karakterisoima graafiongelmaluokka pienenee siirtyessämme luokasta $\mathfrak{M}_{a,b}^c$ luokkaan $\mathfrak{M}_{a,b}$ ja edelleen siirtyessämme luokkaan $\mathfrak{M}_{a,*}$ tai $\mathfrak{M}_{*,b}$.
- (b) MML karakterisoi luokassa $\mathfrak{M}_{a,b}$ saman graafiongelmaluokan kuin GML luokassa $\mathfrak{M}_{a,*}$.
- (c) Logiikan MML luokissa $\mathfrak{M}_{a,*}$ ja $\mathfrak{M}_{*,b}$ karakterisoimat graafiongelmaluokat ovat keskenään vertailemattomia.
- (d) MML karakterisoi luokassa $\mathfrak{M}_{*,b}$ saman graafiongelmaluokan kuin GML luokassa $\mathfrak{M}_{*,*}$.

Todistus. Nämä ovat selviä seurauksia graafiongelmaluokkien karakterisoinnista lauseessa 4.3 ja todistamastamme vakioaikaisten luokkien välisestä järjestyksestä (kuva 3.4). \square

Lähteet

- [1] Benthem, J. V., Eijck, J. V., Stebletsova, V., *Modal Logic, Transition Systems and Processes, Journal of Logic and Computation* 4.5 (1994) 811-855.
- [2] Blackburn, P., de Rijke, M., Venema, Y., *Modal Logic*, Cambridge University Press, Cambridge, 2001.
- [3] Blackburn, P., van Benthem, J., *Modal Logic: A Semantic Perspective, Studies in Logic and Practical Reasoning*, Vol. 3. Elsevier, (2007) 1-84.
- [4] Bondy, J. A., Murty, U. S. R., *Graph theory with applications* Vol. 290. London: Macmillan, 1976.
- [5] De Rijke, M., *A note on graded modal logic, Studia Logica* 64.2 (2000) 271-283.
- [6] Hella, L., Järvisalo, M., Kuusisto, A., Laurinharju, J., Lempiäinen, T., Luosto, K., Suomela, J., Virtema, J., *Weak Models of Distributed Computing, with Connections to Modal Logic, Distributed Computing* 28.1 (2015) 31-53.
- [7] Suomela, J., *Survey of local algorithms, ACM Computing Surveys (CSUR)* 45.2 (2013) 1-40.
- [8] Vasudev, C., *Graph theory with applications*, New Age International, 2006.
- [9] Yamashita, M., Kameda, T., *Computing on anonymous networks. I. Characterizing the solvable cases, IEEE Transactions on Parallel and Distributed Systems* 7.1 (1996) 69-89.