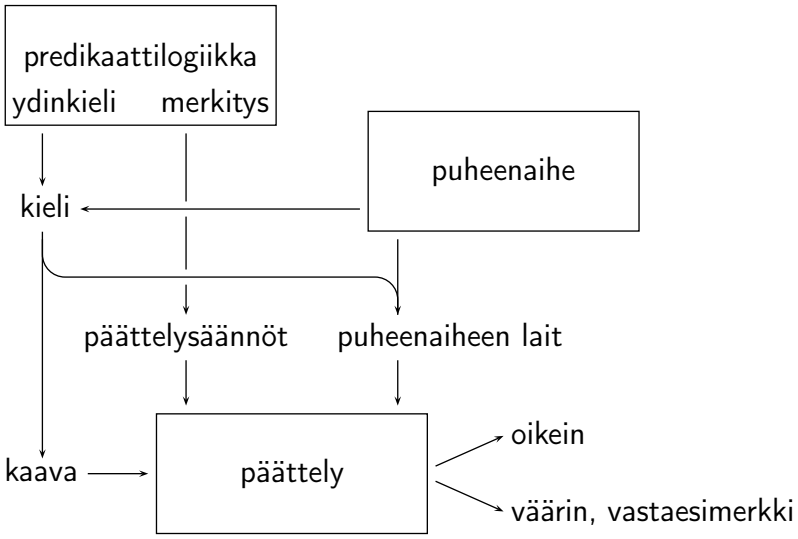
Mistä logiikassa on kyse

Antti Valmari
Jyväskylän yliopisto

19. marraskuuta 2024



Kuva 1: Nykyaikaisen logiikan käsitteiden välisiä suhteita

1 Johdanto

Logiikka on pätevän päättelemisen tiede. Se sai alkunsa yli 2000 vuotta sitten muun muassa Aristoteleen kirjoituksista. Nykyaikaisen logiikan kehityksen katsotaan alkaneen George Boolen 1847 kirjasta ”The Mathematical Analysis of Logic”. Vielä 1930-luvulla logiikka oli melkoisessa myllerryksessä. Nykyaikainen logiikka muistuttaa Aristoteleen logiikkaa vain muutamien peruskäsitteiden tasolla.

Tämä teksti ja siihen liittyvät nettitehtävät esittelevät nykyaikaista logiikkaa, keskittyen suuriin linjoihin ja jättäen tekniset yksityiskohdat vähälle huomiolle. Niissä tutustutaan siihen, miten logiikan kielellä voi puhua asioista, mitä tarkoittaa että väite on tosi ja mitkä ovat pätevän päättelyn säännöt. Tavoitteena on, että kuvan 1 sisältö tulisi vähän kerrassaan ymmärrettäväksi ja tutuksi. Lisäksi kerrotaan, mitä yllättävää 1930-luvulla saatiin selville siitä, mihin logiikka pystyy ja mihin se ei pysty. Myös tarkastellaan muutamia niistä monista yhteyksistä, joita vallitsee toisaalta logiikan ja toisaalta tietokoneiden ja varsinkin niiden ohjelmoinnin välillä.

Tekstiin liittyvät nettitehtävät on kytketty ohjelmaan, joka tarkastaa vastaukset ja antaa niistä palautetta. Ohjelma osaa varsin paljon logiikkaa. Se pystyy tarkastamaan kaavoja ja päättelyitä. Niinpä tehtävät ovat paljon monipuolisempia kuin pelkästään monivalintoja.

Teksti, nettitehtävät ja palauteohjelma on tarkoitettu käytettäviksi vapaaseen itseopiskeluun, suurelle yleisölle tarjottavalla kurssilla taikka laajemman, yliopistollisen kurssin osana. Niiden käyttö ei itsessään vaadi rekisteröitymistä minne-

kään. Jos niitä käytetään jonkin kurssin osana, niin kurssilla on omat rekisteröitymisen ja suoritusten seurannan käytäntönsä.

Jotta tekstiä voisi lukea myös ilman pääsyä nettiin, on osa nettitehtävien vastauksista kerrottu myös tekstissä. Ne eivät kuitenkaan ole tekstissä heti kysymysten jälkeen, jotta ne eivät osuisi ennenaikaisesti heidän silmiinsä, jotka haluavat tehdä nettitehtävät. Niinpä, kun tuntuu siltä, että tietoa nettitehtävien vastauksista puuttuu, kannattaa jatkaa lukemista.

Tekstin sisällä on linkkejä tyyliin ”sivulla 13”. Joissakin PDF-tiedostojen lukuohjelmissa on sellainen kätevä ominaisuus, että kun kursorin siirtää linkin päälle (mutta ei klikkaa), niin linkin takana olevasta tekstistä avautuu heti muutaman rivin mittainen näyte. Linkistä 13 pitäisi avautua katkelma, jossa lukee muun muassa ”Hän ottaa (teetä tai kahvia) ja maitoa”. Tämä ominaisuus tehostaa opiskelua, joten sen käyttöön saamiseksi kannattaa nähdä tämän tekstin omalle koneelle lataamisen ja hyvällä lukuohjelmalla avaamisen vaiva.

Teksti on yritetty laatia siten, että uudet asiat tulevat vähän kerrallaan ja asiayhteyksissä, jotka tekevät ne ymmärrettäviksi. Siitä on seurannut, että isoja teemoja kuten päätteleminen ei ole voitu käsitellä alusta loppuun yhdellä kertaa yhdessä luvussa, vaan ne on hajautettu sinne tänne. Siksi kunkin luvun otsikko kuvastaa luvun pääteemaa, mutta luku saattaa sisältää paljon muutakin asiaa, josta kerrotaan sen verran kuin pääteeman käsittely tarvitsee, ja joka itse käsitellään valmiiksi myöhemmässä luvussa.

Tekstin lopussa on luettelo, jossa on symbolien ja termien selityksiä sekä englanninkielisten termien suomennoksia (sivu 62). Siinä määrin kuin kohtuudella mahdollista, selitykset on pyritty laatimaan ymmärrettäviksi vähillä taustatiedoilla, välttäen matemaattisia merkintöjä. Joihinkin selityksiin sisältyy myös tietoa, joka auttaa kokonaiskuvan muodostamisessa, kuten miksi ensimmäisen kertaluvun logiikka on erityisen tärkeä. Tavoitteena on, että aina kun tässä tekstissä tulee vastaan sana tai symboli jonka merkitys ei vielä ole selvinnyt tai on unohtunut, se kannattaa katsoa luettelosta. Luettelo on sisällöltään laajempi kuin tämä teksti, jotta siitä olisi mahdollisimman paljon hyötyä englanninkielisen aineiston lukemisessa ja logiikan opintojen jatkamisessa.

Englanninkielisen Wikipedian logiikkaa käsittelevät sivut ovat enimmäkseen luotettavia. Niistä löytyy melko helppotajuisiakin esittelyitä tärkeimmille peruskäsitteille. Ikävä kyllä helppotajuisen esittelyn löytäminen on toisinaan työlästä, sillä samaa asiaa saatetaan käsitellä usealla eri sivulla, joista osa jättää yleistajuisen puolen kertomatta. Moni sivu kertoo perustietojen lisäksi monenlaisia erikoistietoja, joten lukijan on osattava tunnistaa missä kohtaa hänen kannattaa lopettaa lukeminen.

Toinen ongelma on, että Wikipedian logiikkasivut eivät ole keskenään yhtenäiset. Muun muassa samasta asiasta saatetaan käyttää eri sivuilla eri merkintö-

jä. Tämä ongelma vaivaa logiikan kirjallisuutta laajasti. Esimerkiksi ”epätosi” on joissakin teksteissä F, joissakin toisissa 0 ja kolmansissa $\bot$.

Netissä on myös runsaasti alkeisopiskeluun tarkoitettuja nettisivuja ja videoita. Niihin kannattaa suhtautua varauksella, sillä niissä saattaa olla paljon virheitä ja mutkien suoraksi vetämisiä. Jotkin asiat saattavat silti aueta niistä paremmin kuin tästä tekstistä ja Wikipediasta. Ainakin forall x: Calgary ja Stanford Introduction to Logic vaikuttavat päteviltä.

Netissä vapaasti luettavissa oleva teos Stanford Encyclopedia of Philosophy sisältää lukuisia päteviä kirjoituksia logiikan eri aiheista. Joukossa on sekä matematiikkaa välttäviä historiallisia katsauksia että vaativia matemaattisia artikkeleita. Moni tämän tekstin yksityiskohta on tarkastettu sieltä.

2 Monenlaisia perusasioita

Tässä luvussa aloitetaan sen kertominen, miten väitteitä voi esittää logiikan merkinnöillä ja miten väitteistä voi päätellä toisia väitteitä. Myös esitellään miten tietokoneiden rakenne perustuu totuusarvoilla laskemiseen. Lisäksi luvun alussa tutustutaan nettitehtävien vastausten kirjoittamiseen ja niiden automaattisen tarkastamisen tuottamaan palautteeseen.

2.1 Palauteohjelman käyttäminen

Kuten johdannossa kerrottiin, tähän tekstiin liittyy nettitehtäviä. Ne on kytketty ohjelmaan nimeltä MathCheck, joka tarkastaa vastaukset ja antaa niistä palautetta. Tässä luvussa tutustumme vastausten kirjoittamiseen ja palautteen tulkintaan. Samalla kohtaamme alustavasti joitakin logiikan perusasioita.

Aluksi on löydettävä ensimmäinen tehtävä netistä. Riippuen laitteesta jolla luet tätä tekstiä, saattaa olla, että se onnistuu klikkaamalla alla olevaa linkkiä:

tässä on linkki ensimmäiseen tehtävään

Jos onnistui, niin avautuu ruutu, jonka vasen reuna näyttää suunnilleen tältä

Kirjoita kaava, joka sanoo, että x on suurempi kuin 5. Vihje

uuteen tai oikealle 1

ja oikean reunan yläosa suunnilleen tältä

Avaa ohje: aritmetiikka vertailut peruslogiikka päättelyt kvanttorit

Jos tehtävä ei auennut linkkiä klikkaamalla, mene tälle veppisivulle

http://users.jyu.fi/%7eava/logiikassa_kyse2.html

ja skrollaa kunnes tehtävä löytyy. Skrollatessa vain vasen puoli liikkuu ja oikea puoli pysyy paikallaan.

Kun olet saanut tehtävän auki, voit seuraavaksi vastata siihen tai katsoa vihjeen. Vaikka et tällä kertaa tarvitsisikaan vihjettä, se kannattaa katsoa ennemmin tai myöhemmin, jotta vihjeiden katsominen tulisi tutuksi jatkoa varten. Vihje ilmestyy näkyviin, kun siirrät kursorin vaaleanruskealla taustalla olevan sanan ”Vihje” päälle.

Vastaaminen tapahtuu klikkaamalla vastauslaatikkoa, kirjoittamalla vastaus sinne ja klikkaamalla jompaa kumpaa vastauslaatikon alla olevista napeista. Tässä vaiheessa kannattaa käyttää nappia ”oikealle”. Tarvitsemme nappia ”uuteen” vasta sivulla 59.

Jos vastasit oikein, oikean puolen vaaleanharmaa laatikko muuttui valkoiseksi, ja siihen ilmestyi esimerkiksi seuraava teksti:

> model-answer $\Leftrightarrow$ $x > 5$
>
> Oikein!

Edellä oli sana ”esimerkiksi”, koska oikeita vastauksia on muitakin kuin $x > 5$. Muun muassa myös $5 < x$ ja $x > 2 + 3$ ovat oikein. MathCheck hyväksyy myös ne. Kokeile, niin näet!

Sen sijaan $X > 5$ ei ole oikein. Se ei ole oikein, koska matematiikassa iso kirjain tarkoittaa melkein aina eri asiaa kuin vastaava pieni kirjain. Kokeile vastausta $X > 5$ ja yritä tulkita MathCheckin antama palaute. Kannattaa tehdä se nyt heti, ennen kuin jatkat lukemista tästä kohdasta eteenpäin! Palaute on selitetty alla, mutta oppimiselle on hyödyksi yrittää tulkita se ensin itse.

MathCheck antaa vastauksesta $X > 5$ palautteeksi punaisen englanninkielisen tekstin, jossa sanotaan, että ”relaatio ei päde kun $x = 5$ ja $X = 6$” sekä ”vasen $\equiv$ F” ja ”oikea $\equiv$ T”. Se tarkoittaa, että jos muuttujan x arvoksi annetaan 5 ja muuttujan X arvoksi annetaan 6, niin $\Leftrightarrow$-symbolin vasen puoli on epätosi mutta oikea puoli on tosi. MathCheck siis tulkitsi symbolit x ja X eri muuttujiksi, kuten matematiikassa on tapana. Palautteen alussa oleva sana ”model-answer” edustaa niin sanottua mallivastausta, eli jotakin oikeaa vastausta. Kirjaimet F ja T tulevat englanninkielen sanoista ”false” ja ”true”.

Tässä tapauksessa mallivastaus on $x > 5$. Kun x saa arvon 5, niin mallivastaus sievenee väitteeksi $5 > 5$ eli ”viisi on suurempi kuin viisi”. Se ei ole totta. Kun X saa arvon 6, niin $X > 5$ sievenee väitteeksi $6 > 5$ eli ”kuusi on suurempi kuin viisi”. Se on totta. Symboli $\Leftrightarrow$ vaatii, että valitaanpa x:n ja X:n arvot miten tahansa sallituista mahdollisuuksista, niin joko molemmat puolet ovat totta tai kumpikaan puoli ei ole totta. Tässä tapauksessa sallittuja valintoja ovat kaikki lukusuoran

luvut. MathCheck kertoi, että tällä kertaa vaatimus rikkoutui, koska valitsemalla x:ksi 5 ja X:ksi 6 tuli vasemmasta puolesta epätosi mutta oikeasta puolesta tosi.

Valinta $x = 5$ ja $X = 6$ ei ole ainoa, jolla toinen kaavoista $x > 5$ ja $X > 5$ on tosi mutta toinen ei ole. Myös valinnoilla $x = 4$ ja $X = 8$ käy niin, ja valinnalla $x = 6$ ja $X = 5$ käy niin. Vaihtoehtoja on loputtomasti. Symbolin $\Leftrightarrow$ tapauksessa mikä tahansa valinta sallituista mahdollisuuksista, jolla toinen puoli on ja toinen puoli ei ole totta, on *vastaesimerkki* (englanniksi *counter-example*). Matematiikassa yksikin vastaesimerkki riittää osoittamaan kohteensa virheelliseksi. Matematiikassa ei päde ”poikkeus vahvistaa säännön”, vaan ”poikkeus kumoaa säännön”.

Nyt on aika siirtyä seuraavaan tehtävään. Voit joko skrollata veppisivua hieman alaspäin tai klikata tätä linkkiä.

Tehtävässä pyydetään kirjoittamaan mahdollisimman lyhyt kaava, joka sanoo, että x on suurempi tai yhtäsuuri kuin 5. Matematiikassa ”suurempi tai yhtäsuuri” voidaan ilmaista symbolilla ”$\geq$”. Mutta tavallisessa tietokoneen näppäimistössä ei ole sellaista merkkiä. Tähän pulmaan on kaksi ratkaisua: yksi joka on helppo muistaa mutta kömpelö käyttää, ja toinen joka on hieman vaikeampi muistaa mutta mukavampi käyttää. Seuraavaksi kokeilemme niitä molempia.

Ensimmäinen ratkaisu on etsiä jostakin ”$\geq$”, maalata ja kopioida se sekä pudottaa vastauslaatikkoon. Se löytyy vaikka tuosta: Maalaamismenetelmä on yhden symbolin tapauksessa toki kömpelö. Mutta se toimii melko usein myös pitempien jaksojen tapauksessa, ja silloin siitä on paljon hyötyä. Siksi sitä kannattaa kokeilla nyt. Kirjoita tehtävän vastaus siten, että tuotat symbolin ”$\geq$” maalaamalla, kopioimalla ja pudottamalla.

Seuraavaksi kokeilemme maalaamismenetelmää monesta merkistä koostuvaan jaksoon. Pyyhi edellinen vastaus pois. Sitten maalaa ja kopioi alla oleva rivi kokonaisuudessaan, pudota se vastaukseksi ja klikkaa oikealle-nappia:

/* Kokeile tätä! */ $x \geq 5$

Jos kaikki sujui kuten piti, niin teksti ”Kokeile tätä!” tuli osaksi MathCheckin antamaa palautetta.

Toinen ratkaisu on kirjoittaa `>=` . Monet ohjelmointikielet ja ohjelmat tulkitsevat sen tarkoittamaan ”$\geq$”, ja siksi sama käytäntö toteutettiin myös MathCheckiin. Jotta tämäkin menetelmä jäisi mieleen, kokeile sitä kirjoittamalla vastauslaatikkoon `x >= 5` ja klikkaamalla oikealle-nappia.

Ehkä huomasit edellisestä esimerkistä, että vastaukseen saa kirjoittaa myös tyhjää. MathCheck ei välitä siitä onko tyhjää paljon vai vähän, mutta tyhjän avulla voi tehdä vastauksen itselle helpommin hahmotettavaksi. Vastauksen voi jopa jakaa monelle riville. Kokeile, äläkä murehdi sitä, että vastauslaatikossa on tilaa vain yhdelle riville:

```
x
>=
5
```

Kuten toivottavasti huomasit, vastaus saa olla useampirivinen kuin mitä vastauslaatikossa näkyy. Jos haluat piilossa olevat rivit näkyviin, niin riippuen siitä mitä veppiselainta käytät, siihen saattaa olla keino. Vastauslaatikon jossakin nurkassa saattaa olla kohta, josta sitä voi suurentaa (ja pienentää) hiirellä vetämällä.

Sitä MathCheck ei kuitenkaan siedä, että yhtä asiaa tarkoittavan merkkijonon sisällä on tyhjää. Kokeile, miten se reagoi seuraavaan vastaukseen (varaudu saamaan sekava palaute): `x > = 5`

Palautteessa oleva punareunainen musta laatikko näyttää kohdan, jossa MathCheck lakkasi ymmärtämästä vastaustasi. Teksti ”Got $=$, but expected ...” kertoo mitä MathCheck siinä kohdassa sai ja luettelee, mitä sen mielestä olisi saanut tulla. Palautteessa ei sanota, että vikana oli välilyönti merkkien `>` ja `=` välissä, koska riippuen siitä mitä vastaaja oli halunnut ilmaista, voi vikana olla jokin muukin. Ihmisten ja tietokoneiden välistä kommunikaatiota on mahdoton saada sellaiseksi, että tietokoneen antamat vastaukset olisivat aina helppotajuisia.

Tässäkin tehtävässä MathCheck hyväksyy, että vertailu kirjoitetaan toisinpäin. Sitä varten täytyy saada ilmaistua symboli ”$\leq$”. Saattaa olla helppo arvata miten sen voi ilmaista tavalisen tietokoneen näppäimistön merkeillä, mutta jatkossa tarvitsemme myös symboleita, joiden tapauksessa arvaaminen on vaikeampaa, kuten ”$\cdot$”, ”$\neq$”, ”$\neg$” ja ”$\equiv$”.

Siksi veppisivun oikean reunan yläosaan on laitettu ohje. Siellä on sanoja vaaleanruskealla taustalla. Kun kursorin siirtää jonkin niistä päälle, niin näkyviin ilmestyy luettelo, jossa kerrotaan, miten vastaavaan aihepiiriin kuuluvia symboleita voi ilmaista näppäimistön merkeillä. Harjoituksen vuoksi, etsi sieltä miten ”$\leq$” voidaan ilmaista näppäimistön merkeillä!

Tämä tehtävä poikkesi edellisestä sikäli, että kirjoitettavan kaavan pitää olla mahdollisimman lyhyt. Siksi MathCheck ei hyväksy vastausta $x \geq 2+3$. Vastaus $x \geq 2+3$ on kuitenkin epäkelpo vain lievemmällä tavalla kuin vastaus $X > 5$ oli aiemmassa tehtävässä. Siellä $X > 5$ oli matemaattisesti väärin, ja MathCheck antoi sille vastaesimerkin. Mutta nyt vastaesimerkkejä ei ole olemassa, sillä aina kun $x \geq 5$ on tosi, myös $x \geq 2+3$ on tosi, ja päinvastoin. Vastaus $x \geq 2+3$ ei siis ole matemaattisesti väärin, vaan se on jollakin muulla tavalla erilainen kuin tehtävän laatija halusi (se on liian pitkä).

Siksi MathCheck antaa erilaisen virheilmoituksen. Ennen kuin jatkat lukemista, kokeile vastausta $x \geq 2+3$ ja yritä tulkita MathCheckin antama palaute!

MathCheckin antama palaute tarkoittaa, että kaavasi oli liian monimutkainen. Siinä oli viisi perusosaa, kun olisi saanut olla enintään kolme. Perusosien määrän

tarkka laskeminen on liian monimutkaista tässä kokonaan kerrottavaksi. Tässä yhteydessä riittää tietää, että melkein jokainen yksittäinen symboli on perusosa, mutta sulkeita ”(” ja ”)” ei lasketa perusosiksi. Niinpä, jos saat palautteen että kaavasi oli liian monimutkainen, niin yritä lyhentää sitä, mutta älä turhaan näe vaivaa sulkeiden poistamiseksi, sillä siitä ei ole apua.

Kaavan muokkaaminen mahdollisimman lyhyeksi ei läheskään aina ole järkevä tavoite. Tavallisesti järkevintä on tavoitella kaavaa, joka sanoo asian mahdollisimman helposti ymmärrettävällä tavalla. Siksi vain harvaan tehtävään on kirjoitettu vaatimus, että vastauksen pitää olla mahdollisimman lyhyt.

Tehtävään saattaa silti olla asetettu monimutkaisuusraja hylkäämään vastaukset, jotka ovat monimutkaisempia kuin hyvät vastaukset. Joihinkin tehtäviin on olemassa erityisen lyhyt vastaus, joka on niin vaikea hoksata että sen hoksaamista ei ole asetettu vaatimukseksi, mutta jonka hoksaamisesta kannattaa kehua. Sitä varten saattaa olla asetettu toinen raja. Seuraava esimerkki havainnollistaa näitä.

Esimerkissä puheenaiheena on kokonaisluvut. Siis esimerkiksi luku 3 on mahdollinen, mutta 3,25 ja $3\frac{1}{4}$ eivät ole. Tavoitteena on kaava, joka sanoo että luku n on lukujen 4 ja 8 välillä, rajat mukaan lukien. Sen voi sanoa kömpelösti sanomalla, että $n = 3$ tai $n = 4$ tai $\ldots$ tai $n = 8$. ”Tai” voidaan logiikassa ilmaista symbolilla ”$\vee$”. Tämän linkin takaa löytyy tehtävä, johon on kirjoitettu valmiiksi tälläinen vastaus. Kokeile, minkä palautteen MathCheck antaa siitä!

Edellä mainitusta vastauksesta $n = 3 \vee n = 4 \vee n = 6 \vee n = 5 \vee n = 7 \vee n = 8$ ei voi olla varma, että mukana ovat kaikki kokonaisluvut neljästä kahdeksaan ja vain ne, ellei lue sitä kokonaan ajatuksen kanssa. Niinpä se on paitsi pitkä, myös työläs tulkita. Vastauksen saa lyhyemmäksi ja helppotajuisemmaksi kirjoittamalla kaavan, joka sanoo, että n on vähintään yhtäsuuri kuin 3 ja enintään yhtäsuuri kuin 8. ”Ja” voidaan logiikassa ilmaista symbolilla ”$\wedge$”. Vaihda vastaus sellaiseksi, ja kokeile minkälainen palaute tulee!

Palaute tarkoittaa, että vastaus kelpaa, mutta lyhyempikin vastaus on olemassa. Sen etsiminen on lisätehtävä heille, joilla riittää intoa! Jos et halua etsiä sitä tai et yrityksistäsi huolimatta keksinyt sitä, niin siirrä kursori vastausnappien alapuolella olevan tekstin ” Hyvin lyhyt oikea vastaus ” päälle.

Tehtävään voi myös olla asetettu kielto käyttää jotakin tiettyä symbolia tai symboleita. Tämän linkin takana pyydetään ilmaisemaan $x \geq 5$ mahdollisimman lyhyesti siten, että symboleita ”$\leq$” ja ”$\geq$” ei saa käyttää. Kokeile ensin, minkälainen palaute tulee, jos kiellosta huolimatta vastaus on $x \geq 5$ (tai $5 \leq x$).

Tehtävässä on lisäksi sanottu, että muuttujan x arvo voi olla vain kokonaisluku. Tämä lisätieto tekee varsin lyhyen vastauksen mahdolliseksi. Koeta keksiä se!

Jos myös kokonaislukujen väliin jäävät luvut kuten $4\frac{1}{2}$ sallitaan, niin edellä löydetty lyhyt vastaus ei ole pätevä. Silti yhä on mahdollista ilmaista $x \geq 5$ käyttämättä symboleita ”$\leq$” ja ”$\geq$”, tarvitaan vain hieman pitempi ilmaus. Se on tehtävänä tämän linkin takana. Kokeile ensin mikä palaute tulee edellisen tehtävän oikeaan vastaukseen $x > 4$, ja sitten yritä keksiä vastaus, joka kelpaa tähän tehtävään!

Tehtävässä käytettiin sanaa ”reaaliluvut”. Se tarkoittaa kokonaislukuja ja kaikkia kokonaislukujen välissä olevia lukuja. Kaikki murtoluvut ovat reaalilukuja. Myös π eli ympyrän kehän ja halkaisijan suhde on reaaliluku, vaikka se ei ole murtoluku. On olemassa myös lukuja, jotka eivät ole kokonaislukuja eivätkä sijaitse kokonaislukujen välissä, mutta niitä ei tässä tekstissä käsitellä.

Edellisen tehtävän vastaukseksi kelpaa $x > 5 \vee x = 5$. Osien järjestyksellä ja esittämissuunnilla ei ole väliä, kunhan suuntaa vaihdettaessa ”$>$” käännetään toisinpäin. Niinpä muun muassa myös $5 = x \vee x > 5$ ja $x = 5 \vee 5 < x$ ovat oikein.

2.2 Tosi, epätosi, ja, tai, ei, ”(” ja ”)”

Edellisessä luvussa kohtasimme alustavasti symbolit ”$\wedge$”, ”$\vee$”, ”T”, ”F” ja ”$\Leftrightarrow$”, jotka vastaavat varsin pitkälle luonnollisen kielen käsitteitä ”ja”, ”tai”, ”tosi”, ”epätosi” ja ”on tosi yhtäaikaa kun”. Tässä luvussa paneudumme perusteellisemmin niistä muihin kuin ”$\Leftrightarrow$”. Opettelemme sanomaan myös ”ei”. Harjoittelemme luonnollisen kielen ilmauksien kääntämistä kaavoiksi. Symboliin ”$\Leftrightarrow$” tutustumista jatkamme luvuissa 2.3 ja 5.

Logiikassa ”ei” voidaan ilmaista symbolilla ”$\neg$”. Se kirjoitetaan kohteensa eteen. Vastauslaatikkoon voi sitä edustamaan kirjoittaa `!` . Saattaa olla tarpeen kirjoittaa kohteen ympärille sulkeet ”(” ja ”)”. Ne saa jättää pois silloin, kun ilman niitäkin on helppo huomata, kuinka pitkälle kohde ulottuu.

Käyttämättä symbolia ”F”, kirjoita tänne mahdollisimman lyhyt kaava, joka sanoo ”epätosi”! Sitten kirjoita hieman alempana olevaan vastauslaatikkoon ”epätosi” vielä rajoitetummalla joukolla symboleita!

Jälkimmäiseen tehtävään on olemassa useita vaaditun lyhyitä vastauksia. Keksitkö sellaisen, jossa ei esiinny yhtään muuttujaa? Entä keksitkö sellaisen, jossa esiintyy muuttuja? Riippumatta siitä, keksitkö, jatka lukemista. Kokeilemme muutamia liian pitkiä vastauksia, koska se antaa lisää kokemusta siitä, miten erilaisilla kaavoilla voidaan sanoa sama asia.

Esimerkki (liian pitkästä) kaavasta, jossa esiintyy muuttuja mutta joka ei ole tosi millään sen arvolla, on $0x = 1$. Kokeile myös sitä äskeiseen vastauslaatikkoon. Ehkä jo muistatkin, miten palaute tulkitaan, mutta siltä varalta että et vielä muista: saatu palaute tarkoittaa, että vastaus on matemaattisesti oikein, mutta liian monimutkainen. (Siltä varalta, että huomasit, että palautteen mukaan monimutkaisuus

oli 5 eikä 4: kertolasku laskettiin perusosaksi, vaikka se esitettiin ilman omaa symbolia kirjoittamalla 0 ja x välittömästi peräkkäin.) Klikkaa ”Monimutkaisuusraja on käytössä” pois päältä ja klikkaa uudelleen oikealle-nappia!

Kokeile myös kaavoja $1x = 1$ ja $0x = 0$. Ne eivät ole matemaattisesti oikeita vastauksia tehtävään, joten niistä tulevissa palautteissa on iso ero edelliseen. Koska tehtävänä oli kirjoittaa kaava, joka on aina epätosi, ovat palautteissa annettavat vastaesimerkit ikään kuin nurinkurin: ne kertovat muuttujan x arvon, jolla kaava on tosi! Kaavalle $0x = 0$ siihen tehtävään kelpaa mikä tahansa luku, mutta kaava $1x = 1$ ei ole epätosi vain yhdellä x:n arvolla, nimittäin 1.

Päättele mielessäsi, kelpaisivatko $\neg(0x = 0)$ ja $\neg(1x = 1)$ tehtävän vastauksiksi, ja sitten kokeile ennustitko oikein.

Esimerkiksi $0 = 1$ on tarpeeksi lyhyt muuttujaton vastaus. Esimerkiksi $x \neq x$ on tarpeeksi lyhyt muuttujallinen vastaus.

Logiikan näkökulmasta ei ole olennaista eroa siinä, ilmaistaanko ”epätosi” symbolilla ”F” vai jollakin kaavalla, joka ei ole tosi millään siinä esiintyvien muuttujien arvojen valinnoilla. Tämä voi tuntua hassulta! Syy siihen, miksi logiikka toimii näin, vaatii enemmän taustatietoja kuin tähän mennessä on käsitelty. Asiaan tullaan palaamaan sivulla 59.

Jos kaavassa ei ole muuttujia, niin se on joko aina tosi tai aina epätosi. Onko se aina tosi vai aina epätosi, saattaa riippua puheenaiheesta. Jos kaavassa on muuttujia, niin mukaan tulee kolmas mahdollisuus: kaavan totuusarvo riippuu niille annettavista arvoista. Vaikka kaavassa olisi muuttujia, niin se saattaa silti olla aina tosi (tai aina epätosi). Kun puheenaihetta vaihdetaan, niin kaava saattaa siirtyä mahdollisuudesta toiseen. Kokeile sitä täällä! Siellä puheenaiheeksi voi valita joko kokonaisluvut tai aritmetiikan modulo 10. Jälkimmäinen tarkoittaa suunnilleen sitä, että luvuista otetaan huomioon vain viimeinen numero.

(Kokeilemalla kaavat $-2 = 2$ ja $-2 = 8$ tai kaavat $\frac{6}{3} = 2$ ja $\frac{6}{2} = 3$ huomaat, että asia ei ole ihan näin yksinkertainen. Mutta nyt tavoitteena ei ole tutustua aritmetiikkaan modulo 10 yksityiskohtaisesti, joten emme jatka tähän suuntaan.)

Kun puheenaihetta vaihdetaan, niin kaava voi jopa lakata olemasta kaava. Tämä johtuu siitä, että eri puheenaiheilla on käytössä eri symboleita. Kokeile edellisen tehtävän vastauslaatikossa kaavaa $\sqrt{6} = 4$ molemmilla puheenaiheilla.

Koska kokonaisluvut ajatellaan usein osajoukoksi reaaliluvuista (eikä itsenäiseksi teoriaksi), ja koska useimpien kokonaislukujen neliöjuuret eivät ole kokonaislukuja, on parempi jättää neliöjuuri määrittelemättä silloin, kun kokonaisluvut ovat mukana mutta muut reaaliluvut eivät ole. Sama syy ei päde aritmetiikassa modulo 10, koska se käyttäytyy joka tapauksessa erilailla kuin kokonaisluvut. Neliöjuuri voidaan määritellä siellä mielekkäästi useimmille luvuille (vaikka ei kaikille).

Jos puheenaiheena on kahvitauko, niin numerot, ”$+$”, ”$\leq$” ja niin edelleen eivät ole käytettävissä. Sen sijaan käytettävissä on K tarkoittamaan että hän ottaa

kahvia, T tarkoittamaan että hän ottaa teetä ja niin edelleen. Näilläkin keinoin voidaan muodostaa aina epätosi kaava käyttämättä symboleita F ja T. Kirjoita sellainen kaava tehtäväryhmän viimeiseen vastauslaatikkoon.

Edellisessä luvussa huomasimme, että $x > 5 \vee x = 5$ tarkoittaa samaa kuin $x \geq 5$. Nyt hyödynnämme sitä, että myös ”x ei ole pienempi kuin 5” tarkoittaa samaa kuin $x \geq 5$. Kirjoita ”x ei ole pienempi kuin 5” tämän linkin takaa löytyvään vastauslaatikkoon.

Sanaa ”tai” käytetään luonnollisessa kielessä kahdella eri tavalla. Seuraavat esimerkit havainnollistavat tätä.

> Ravintolassa on eräänä vaihtoehtona kolmen ruokalajin menu. Alkuruoan kohdalla siinä lukee ”päivän keitto tai talon salaatti”. Se tarkoittaa, että asiakkaalla on kaksi vaihtoehtoa: keitto ilman salaattia tai salaatti ilman keittoa. Niinpä tässä tapauksessa ”tai” tarkoittaa, että ”jompikumpi *mutta ei molemmat*”.
>
> Ystävykset suunnittelevat lähtevänsä sunnuntaina uimaan. Toinen sanoo: ”jos sataa tai on kovin kylmä, niin ei lähdetä.” Uintireissu siis perutaan kolmessa eri tapauksessa: sataa mutta ei ole kylmä, on kylmä mutta ei sada, sataa ja on kylmä. Niinpä tässä tapauksessa ”tai” tarkoittaa, että ”jompikumpi *tai molemmat*”.

Eräs logiikan historian kuuluisista kiistoista liittyy tähän eroon. George Boole (1815–1864) oli avainhenkilö logiikan mullistuksessa antiikin ajan ajattelutavoista nykyaikaiseksi. Hänen merkityksensä näkyy muun muassa siinä, että totuusarvoilla laskemista kutsutaan nykyisin Boolen algebraksi, ja monissa ohjelmointikielissä sitä varten on tietotyyppi nimeltä ”boolean” tai ”bool”. Nykyaikaisen logiikan ”tai” on kuitenkin erilainen kuin Boolen versio. Nykyaikaisen version esitti William Stanley Jevons (1835–1882). He kävivät vuosina 1863–1864 kirjeenvaihdon, jossa Jevons yritti turhaan saada Boolen nykyaikaisen ”tai”:n kannalle.

Boolen mukaan täytyy jättää määrittelemättä, mikä on tulos silloin, kun ”tai”:n molemmilla puolilla olevat vaihtoehdot toteutuvat yhtäaikaa. Nykyisin ollaan sitä mieltä, että tai-muotoinen väite voidaan määritellä järkevästi kahdella vaihtoehtoisella tavalla: *poissulkeva tai* (*exclusive or*), jossa väite on tosi täsmälleen silloin kun sen jompikumpi puoli mutta ei molemmat on tosi, ja *sisällyttävä tai* (*inclusive or*), jossa väite on tosi täsmälleen silloin kun sen jompikumpi tai molemmat puolet on tosi.

Kaikki tähänastiset ”$\vee$”:n käyttömme ovat olleet siten valittuja, että tällä erolla ei ole väliä. Mutta nyt on tullut aika tehdä selväksi, kumpaa näistä kahdesta ”$\vee$”

tarkoittaa. Tämän linkin takana on seitsemän tehtävää, jotka havainnollistavat näiden kahden vaihtoehdon eroa ja joista selviää, kumman niistä nykyaikainen ”$\vee$” toteuttaa.

Seuraavissa tehtävissä harjoitellaan kaavojen kirjoittamista käyttäen symboleita ”$\neg$”, ”$\wedge$” ja ”$\vee$”, sulkeita ”(” ja ”)”, sekä seuraavia *propositiomuuttujia* eli muuttujia, joiden arvo on totuusarvo T tai F:

muuttuja	kirjoittaminen	merkitys
K	`K`	hän ottaa kahvia
T	`T`	hän ottaa teetä
M	`M`	hän ottaa maitoa (kahviinsa tai teehensä)
S	`S`	hän ottaa sokeria (kahviinsa tai teehensä)

Tässä on alkajaisiksi kuusi tehtävää, joissa ei tarvita sulkeita.

Ennen tehtäviä, joissa tarvitaan sulkeita, tutustumme vitsiin, jonka tieteiskirjailija Ursula K. Le Guin teki kuuluisaksi. Säveltäjä Todd Barton oli kertonut sen hänelle vuoden 1996 tienoilla. Se kertoo pandasta ravintolassa. Syötyään panda vetäisi esiin aseen, ampui laukauksen ja poistui. Tarjoilija juoksi sen kiinni ja kysyi, miksi se ampui. Panda näytti eläintieteen kirjaa, jossa luki ”Panda. Large black-and-white bear-like mammal, native to China. Eats, shoots and leaves.” Vitsi on siinä, että ”eats shoots and leaves” tarkoittaa ”syö versoja ja lehtiä”, mutta ”eats, shoots and leaves” tarkoittaa ”syö, ampuu ja lähtee”.

Suomenkielisessä virkkeessä ”illalla potilas juo virtsaa ja käy nukkumaan” tapahtuu samankaltainen ilmiö. Pandaesimerkissä väärä tulkinta aiheutui ylimääräisestä pilkusta, mutta tässä esimerkissä se aiheutuu siitä, että pilkku puuttuu. Tarkoitettiin tietenkin, että ”illalla potilas juo, virtsaa ja käy nukkumaan”. Nämä esimerkit havainnollistavat, että ilmauksen rakenteella on toisinaan ratkaiseva vaikutus ilmauksen merkitykseen, vaikka sanat ja niiden järjestys säilyvät samoina.

Virkkeen ”hän ottaa teetä tai kahvia ja maitoa” rakenne on epäselvä. Selvää on, että hän ottaa maitoa ainakin kahvin kanssa, mutta epäselväksi jää, ottaako hän maitoa teen kanssa. Ilmaukset ”hän ottaa teetä tai kahvia, ja maitoa” ja ”hän ottaa teetä, tai kahvia ja maitoa” ovat selkeämpiä (mutta eivät ehkä hyvää suomen kieltä).

Matematiikassa ja logiikassa on tapana poistaa tällaiset epäselvyydet lisäämällä sulkeita. ”Hän ottaa (teetä tai kahvia) ja maitoa” takaa, että hän ottaa maitoa joka tapauksessa, mutta ”hän ottaa teetä tai (kahvia ja maitoa)” sallii, että hän ottaa teetä ilman maitoa.

Kuvassa 2 on esitetty kaikki mahdolliset teen, maidon ja kahvin yhdistelmät kummallakin tulkinnalla. Koska logiikan ”tai” on sisällyttävä, molemmat tulkinnat sallivat, että hän ottaa kaikkia kolmea.

teetä	kahvia	maitoa	(teetä tai kahvia) ja maitoa	teetä tai (kahvia ja maitoa)
ei	ei	ei		
ei	ei	kyllä		
ei	kyllä	ei		
ei	kyllä	kyllä	mahdollinen	mahdollinen
kyllä	ei	ei		mahdollinen
kyllä	ei	kyllä	mahdollinen	mahdollinen
kyllä	kyllä	ei		mahdollinen
kyllä	kyllä	kyllä	mahdollinen	mahdollinen

Kuva 2: ”Hän ottaa teetä tai kahvia ja maitoa” tulkittuna kahdella eri tavalla

Aiemmassa tehtävässä tarvittiin sulkeet ilmauksessa $\neg(x < 5)$, koska ilmauksesta $\neg\ x < 5$ ei näy selvästi, onko symbolin ”$\neg$” kohde pelkkä ”x”, ”$x <$” vai ”$x < 5$”. Se on kyllä pääteltävissä siitä, että x tuottaa luvun, ”$<$” tuottaa totuusarvon ja ”$<$” tarvitsee jotakin molemmille puolilleen, mutta on haluttu, että kaavan rakenne selviää vaivattomammin. Sitäpaitsi ellei erikseen käsketä toisin, ladontaohjelmat eivät tuota $\neg\ x < 5$ vaan vielä vaikeampitajuisen muodon $\neg x < 5$.

Sulkeiden käyttäminen kaavan oikean tulkinnan varmistamiseksi on sallittua melkein aina, vaikka siitä aiheutuisi turhia sulkeita.[1] MathCheckissä tämä on otettu huomioon siten, että se ei piittaa sulkeista lainkaan laskiessaan kaavan monimutkaisuutta. Tuottamastaan palautteesta se jättää tarpeettomat sulkeet pois. Niinpä, jos pyydetään mahdollisimman lyhyttä kaavaa ja oikea vastaus on $x \geq 5$, niin MathCheck hyväksyy oikeiksi vastauksiksi myös muun muassa $x \geq (5)$ ja $(((x) \geq (((5)))))$, ja näyttää ne kaikki muodossa $x \geq 5$. Vastauksen $\neg(((x) < (5)))$ MathCheck näyttää $\neg(x < 5)$.

Toisaalta suuri määrä sulkeita on omiaan vaikeuttamaan kaavojen lukemista. Siksi on sovittu sääntöjä, jotka sallivat jättää sulkeita pois. Jos symbolin ”$\neg$” kohteena on pelkkä muuttuja, pelkkä totuusarvo taikka ilmaus, joka alkaa symbolilla ”$\neg$” tai luvussa 3 esiteltävillä ”$\forall$” tai ”$\exists$”, niin sen ympärillä ei tarvita sulkeita. Siis ei tarvitse kirjoittaa vaikkapa $\neg(\neg(S))$, vaan riittää kirjoittaa $\neg\neg S$.

Ikävä kyllä tämä on melkein ainoa logiikan suljesääntö, jota jokseenkin kaikki kirjoittajat noudattavat. MathCheckissä sulkeet saa jättää pois myös kaavoista muotoa ”hän ottaa teetä tai (kahvia ja maitoa)”, mutta ei saa jättää pois kaavoista muotoa ”hän ottaa (teetä tai kahvia) ja maitoa”. Monet kirjoittajat noudattavat tätä sääntöä, mutta eivät kaikki.

Tässä on neljä tehtävää. Niistä osassa tarvitaan sulkeet. Mutta kaikkiin saa laittaa sulkeet, jos haluaa.

Sanallista ilmausta ei aina saa käännettyä kaavaksi sellaisenaan. Esimerkiksi

[1] Pitkälle menevissä opinnoissa voi kohdata poikkeuksia. Esimerkki: reaaliluvuilla $f^4(x)$ tarkoittaa joko $f(f(f(f(x))))$ tai $(f(x))^4$, mutta $f^{(4)}(x)$ tarkoittaa f:n neljättä derivaattaa.

edellä tehtävässä ”hän ottaa kaikkea” jouduttiin sanan ”kaikkea” tilalle kirjoittamaan osakaava, joka vastaa ilmausta ”kahvia, teetä, maitoa ja sokeria”. Siksi on tarpeen kyetä miettimään erilaisia tapoja ilmaista sama merkitys. Yksi keino harjoitella sitä on koettaa löytää erilaisia kaavoja, joilla on sama merkitys. Tällaisista harjoituksista on toinenkin hyöty: ne auttavat ymmärtämään myöhemmin esitettäviä loogisen päättelyn lakeja.

Täällä on muutama tehtävä, joissa käsketään esittämään väittämiä mahdollisimman lyhyinä kaavoina. (Kannattaa tiedostaa, että tämäkin harjoittelu on aivojumpan vuoksi eikä siksi, että kaavojen saaminen mahdollisimman lyhyiksi olisi muka ihanne. Logiikan sovelluksissa ihanteena ei tavallisesti ole mahdollisimman lyhyt vaan mahdollisimman helppotajuinen kaava.)

Tämän luvun lopuksi harjoittelemme moniosaisten kaavojen muodostamista sanallisten kuvausten perusteella.

Lukujen a ja b *minimi* on pienempi luvuista a ja b. Jos a ja b ovat yhtäsuuret, niin minimi on niiden kanssa yhtäsuuri. Tätä kuvausta ei voi aivan sellaisenaan kääntää kaavaksi, mutta kaavana voi sanoa, että x on jompikumpi luvuista a ja b, ja x on enintään yhtäsuuri kuin kumpikaan niistä. Mene tänne ja rakenna siellä vaiheittain kaava, joka sanoo niin!

Luvun a itseisarvo $|a|$ on a, jos $a \geq 0$, ja $-a$, jos $a < 0$. Käyttämättä symbolia $|\ldots|$, kirjoita tänne kaava, joka sanoo, että $x = |a|$!

2.3 Jonkin verran päättelemisestä

Tässä luvussa aloitamme loogiseen päättelemiseen tutustumisen tutkimalla tärkeimpiä symbolien ”$\neg$”, ”$\wedge$” ja ”$\vee$” $\Leftrightarrow$-muotoisia lakeja. Kokeilemme kolmea tapaa tarkastaa tai todistaa niitä, tutustuen samalla joihinkin laajasti käyttökelpoisiin päättelykeinoihin.

Ennen kuin menemme logiikan piiriin kuuluviin päättelykeinoihin, harjoittelemme MathCheckin antaman palautteen hyödyntämistä. Sekin kehittää ongelmanratkaisutaitoa. Se voi myös auttaa pääsemään myöhemmissä tehtävissä alkuun tai eteenpäin, ja siten välillisesti edistää logiikan oppimista.

Joskus tehtävässä ei pääse alkuun, koska ei hoksaa, mitkä ovat tehtävän muuttujat. Silloin voi olla avuksi keino, jota harjoitellaan täällä parilla tehtävällä: vastaa T (tai F), niin palaute näyttää vastaesimerkin (paitsi jos vastaus olikin oikea). Muuttujat voi tunnistaa siitä.

Niiden jatkeena on pari tehtävää, joissa hyödynnetään MathCheckin antamaa palautetta kaavan monimutkaisuudesta. Siinä kikassa on se huono puoli, että se saattaa johtaa oikeaan vastaukseen ilman, että syntyy ymmärrystä, miksi se on oikea. Tämänkertaisissa harjoituksissa ei ymmärryksen syntyminen ole tavoitteena,

vaan kikkaan tutustuminen. Mutta jos jatkossa päädyt oikeaan vastaukseen pelkästään monimutkaisuuden perusteella, kannattaa jonkin aikaa miettiä, miksi se on oikea.

Eräs jo käsitelty asia on niin tärkeä, että sitä on syytä harjoitella hieman lisää täällä olevalla ryhmällä tehtäviä.

Symbolit "$\neg$", "$\wedge$" ja "$\vee$" noudattavat useita lakeja. Emme käy niitä kaikkia läpi, vaan ainoastaan joitakin hyödyllisimpiä. Osa laeista on helppo keksiä itse, kun muistaa seuraavat periaatteet:

- $\neg$ *kaava* on tosi jos ja vain jos *kaava* on epätosi.
- *kaava*1 $\wedge$ *kaava*2 on tosi jos ja vain jos sekä *kaava*1 että *kaava*2 on tosi.
- *kaava*1 $\vee$ *kaava*2 on tosi jos ja vain jos *kaava*1 tai *kaava*2 tai molemmat on tosi.
- T on aina tosi ja F on aina epätosi.

Täällä olevilla tehtävillä voit harjoitella näitä periaatteita.

Merkintää *kaava*1 $\Leftrightarrow$ *kaava*2 kutsutaan *yhtäpitävyydeksi*.[2] Kuten sivulla 6 kerrottiin, *kaava*1 $\Leftrightarrow$ *kaava*2 tarkoittaa, että jokaisessa mahdollisessa tilanteessa, jossa *kaava*1 on tosi, myös *kaava*2 on tosi, ja päinvastoin. "Tilanne" tarkoittaa kaavoissa esiintyvien muuttujien arvojen yhdistelmää. Tilanne on mahdollinen, ellei erikseen ole sanottu jotakin, joka kieltää sen.

Tilanteiden jako mahdollisiin ja mahdottomiin aiheutuu siitä, että päättelyitä tehdään usein oletusten alaisina. Päättely saatetaan esimerkiksi jakaa kahteen osaan, joista ensimmäisessä tarkastellaan sitä mahdollisuutta, että varas murtautui liikkeeseen yöllä, ja toisessa sitä, että varas saapui liikkeeseen edellisenä iltana sen ollessa vielä auki ja kätkeytyi liikkeen sisään. Koska murtojälkiä ei näy, voidaan ensimmäisessä tapauksessa karsia pois ne epäillyt, jotka eivät osaa murtautua taitavasti. Koska liikkeen sisätilat ovat ahtaat ja myyjä kiersi ne läpi ennen kuin lähti kotiinsa, voidaan jälkimmäisessä tapauksessa karsia pois ne epäillyt, jotka ovat liian isokokoisia mahtuakseen kätkeytymään.

[2] Symbolia "$\Leftrightarrow$" kutsutaan myös loogiseksi ekvivalenssiksi (logical equivalence). Tässä tekstissä tätä vältetään, koska asiaan liittyy sekaannus. Jos määritelmät luetaan tarkasti, niin paljastuu, että toisin kuin toisinaan luullaan, esimerkiksi $x = 1 \Leftrightarrow x + 1 = 2$ ei ole looginen ekvivalenssi, vaan vain matemaattinen totuus. Syy on siinä, että sen pätevyys riippuu symbolien "1", "2" ja "+" tulkinnasta. Käytännön sovelluksissa kuitenkin yleensä oletetaan, että tutuilla symboleilla on tutut merkityksensä, eikä että esimerkiksi "2" tarkoittaakin yhtäkkiä kuutosta. Niinpä symboli "$\Leftrightarrow$" menettäisi suuren osan hyödyllisyydestään, jos sen vaadittaisiin tarkoittavan loogista ekvivalenssia. Siksi tässä tekstissä sitä käytetään tarkoittamaan että molemmat puolet ovat totta tai kumpikaan puoli ei ole totta, olettaen että vakiintuneilla symboleilla on vakiintuneet merkityksensä.

Tässä luvussa voidaan ajatella, että $kaava1 \Leftrightarrow kaava2$ ilmaisee, että kun rajoitutaan mahdollisiin tilanteisiin, niin $kaava1$ ja $kaava2$ tarkoittavat samaa. Tulemme huomaamaan sivulla 34, että asia on tätä monimutkaisempi, mutta emme murehdi sitä vielä nyt.

Moni tärkeä symbolien ”$\neg$”, ”$\wedge$” ja ”$\vee$” laki on ilmaistavissa kaikissa tilanteissa voimassa olevana yhtäpitävyytenä. Itse asiassa edellisessä tehtävässä tehtiin niin. Siellä K tarkoitti ”hän ottaa kahvia”, mutta siellä saatu yhtäpitävyys $\neg\neg K \Leftrightarrow K$ säilyy pätevänä, vaikka K:n tilalle laitettaisiin mikä tahansa kaava. Siis $\neg\neg kaava \Leftrightarrow kaava$, eli jokaiselle kaavalle pätee, että ”ei ei *kaava*” on tosi täsmälleen silloin kun *kaava* on tosi.

Sanan ”*kaava*” ja sitä täydentävien numeroiden käyttö lakien ilmaisemiseen johtaa toisinaan pitkiin ilmauksiin, kuten esimerkiksi seuraava:

$$kaava1 \wedge (kaava2 \vee kaava3) \;\Leftrightarrow\; kaava1 \wedge kaava2 \vee kaava1 \wedge kaava3$$

Myöskään MathCheck ei ymmärrä niitä. Siksi käytämme toisinaan kirjaimia P, Q ja R tarkoittamaan mitä tahansa kaavoja. Niillä edellinen lyhenee seuraavaan muotoon:

$$P \wedge (Q \vee R) \;\Leftrightarrow\; P \wedge Q \vee P \wedge R$$

Tästä aiheutuu lievä sekaannuksen vaara esimerkiksi silloin, kun haluaisimme P:n tarkoittavan ”hän ottaa pullaa”. Siksi logiikassa on tapana käyttää tiettyjä kreikkalaisia kirjaimia edustamaan mitä tahansa kaavoja. Tässä tekstissä kuitenkin vältämme kreikkalaisia kirjaimia (ja yritämme pärjätä ilman pullaa).

MathCheck ei tunne mielivaltaisen kaavan käsitettä, joten sille P, Q ja R ovat totuusarvoisia muuttujia samaan tapaan kuin K. Mutta tehtävät on laadittu siten, että tästä ei ole haittaa.

Edellisen tehtäväryhmän tehtävät kertovat myös, että ”mikä tahansa kaava ja epätosi” on epätosi ja ”mikä tahansa kaava ja tosi” tuottaa saman totuusarvon kuin ”mikä tahansa kaava”. Siis $P \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$ ja $P \wedge \mathsf{T} \Leftrightarrow P$, eli $kaava \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$ ja $kaava \wedge \mathsf{T} \Leftrightarrow kaava$. ”Tai”:lle pätee $P \vee \mathsf{F} \Leftrightarrow P$ ja $P \vee \mathsf{T} \Leftrightarrow \mathsf{T}$. Samat pätevät myös toisinpäin. Esimerkiksi $\mathsf{T} \wedge P \Leftrightarrow P$. Vaikka nämä pikku lait ovat hyvin yksinkertaisia, niitä ei pidä vähätellä! Kohta huomaamme, että niistä on paljon hyötyä.

Kun kaavaa edustavan symbolin tilalle laitetaan kaava, saattaa olla tarpeen lisätä sulkeet varmistamaan, että lisätty kaava säilyy yhtenäisenä kokonaisuutena. Esimerkiksi jos A tarkoittaa että aurinko paistaa ja H tarkoittaa että tuuli humisee hiljaa, ja jos *kaava*:n tilalle ilmauksessa $\neg kaava$ laitetaan $A \wedge H$, niin täytyy kirjoittaa $\neg(A \wedge H)$ eikä $\neg A \wedge H$. Ne eivät tarkoita samaa, sillä ne tuottavat eri totuusarvot silloin kun tuuli ei humise hiljaa. Nimittäin silloin ”aurinko paistaa ja tuuli humisee hiljaa” on epätosi, joten ”ei ole niin, että aurinko paistaa ja tuuli humisee hiljaa” on tosi, mutta ”aurinko ei paista ja tuuli humisee hiljaa” on epätosi. Täällä voit kokeilla, miten MathCheck kertoo, että ne eivät tarkoita samaa.

Monimutkaisten väitteiden tai päättelyiden esittäminen luonnollisella kielellä on usein kömpelöä ja virhealtista. Siksi seuraavaksi opettelemme esittämään niitä logiikan symbolien avulla.

Sen, että tuuli ei humise hiljaa, voi kaavoissa ottaa huomioon sijoittamalla H:n tilalle F. Tässä tapauksessa mahdollisia tilanteita eivät ole kaikki A:n ja H:n totuusarvojen yhdistelmät, vaan vain ne, joissa H on epätosi. Kun tuuli ei humise hiljaa, voimme siis päätellä $\neg(A \wedge H) \Leftrightarrow \neg(A \wedge \mathsf{F})$.

Koska $kaava \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$ pätee mille tahansa *kaava*:lle, se pätee myös kun *kaava*:n tilalla on A. Siksi $A \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$. Logiikassa vallitsee varsin laajasti periaate, että jos kaksi eri ilmausta tarkoittaa samaa, niin ne saa vaihtaa toisikseen isomman kokonaisuuden sisällä. Asia ei ole ihan näin yksinkertainen, mutta sen kiemurat eivät vaikuta tässä luvussa. Siksi käsittelemme kiemuroita vasta sivulla 38. Tässä luvussa tätä periaatetta saa soveltaa yhtäpitävyyteen. Siis

> Sivulla 38 mainittavin rajoituksin, jos $kaava1 \Leftrightarrow kaava2$, niin $isompi\_kaava(kaava1) \Leftrightarrow isompi\_kaava(kaava2)$.

Siksi siitä, että $A \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$, saa päätellä $\neg(A \wedge \mathsf{F}) \Leftrightarrow \neg(\mathsf{F})$. Koska sivun 14 mukaan oikealla puolella ei tarvita sulkeita, voimme kirjoittaa $\neg(A \wedge \mathsf{F}) \Leftrightarrow \neg\mathsf{F}$.

Koska ”¬” tarkoittaa ”ei” ja ”F” tarkoittaa epätosi, ”¬F” tarkoittaa ”ei epätosi”. Mutta sehän on sama kuin tosi! Siksi $\neg\mathsf{F} \Leftrightarrow \mathsf{T}$.

Tällaisen päättelyn saa esittää ketjuna muotoa $kaava1 \Leftrightarrow kaava2 \Leftrightarrow \ldots \Leftrightarrow viimeinen\_kaava$. Olemme tähän mennessä päätelleet

$$\neg(A \wedge H) \Leftrightarrow \neg(A \wedge \mathsf{F}) \Leftrightarrow \neg\mathsf{F} \Leftrightarrow \mathsf{T}$$

Kaavasta $\neg A \wedge H$ saamme samalla tavalla

$$\neg A \wedge H \Leftrightarrow \neg A \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$$

Kumpikin näistä päättelyketjuista nojautuu oletukseen, että tuuli ei humise hiljaa. Kumpikaan ei ota kantaa tilanteisiin, joissa tuuli humisee hiljaa.

MathCheck kykenee tarkastamaan tällaisia päättelyitä. Maalaa, kopioi ja pudota ensimmäinen niistä tänne ja klikkaa oikealle-nappia! Jos kaikki sujui kuten piti, MathCheck hyväksyi sen.

Tehtävässä lukeva `assume ¬H;` määrää, että päättely tarkastetaan oletuksen $\neg H$ alaisuudessa. Oletus $\neg H$ tarkoittaa, että $\neg H$ on tosi eli että H on epätosi.

Osuuden `assume ¬F ∧ T;` selitys on seuraava. Kaavojen maalaaminen, kopiointi ja pudottaminen ei säilytä symbolien F ja F eroa, vaan muuttaa symbolin F eli ”epätosi” muuttujaksi F, joka voi olla sekä tosi että epätosi. Se tekee saman myös symboleille T ja T. Tämä ongelma kierrettiin lisäämällä oletuksiin $\neg F$ ja T, eli että F voi olla vain epätosi ja T voi olla vain tosi. Niinpä F ja T kelpaavat F:n ja T:n tilalle, vaikka F ja T ovatkin muuttujia eivätkä totuusarvoja.

Sitten lisää päättelyn perään "⇔" ja jälkimmäinen päättely. Siten syntyy yksi pitkä, epäpätevä päättely. Klikkaa oikealle-nappia uudelleen! MathCheck kertoo, että päättelyssä on virhe lisätyn "⇔":n kohdalla, ja antaa vastaesimerkin eli esimerkin muuttujien arvoista, joilla virhe ilmenee. Vastaesimerkissä H ja F ovat vääjäämättä epätodet ja T tosi, koska oletukset eivät salli muuta. Tässä tapauksessa mikään ei määrää kumpi totuusarvo A:lla pitää olla, sillä vastaesimerkki toimii molemmilla vaihtoehdoilla. Niinpä MathCheck ilmoitti joko $A \equiv \mathsf{F}$ tai $A \equiv \mathsf{T}$.

Seuraavaksi pyyhi ensimmäinen päättely ja väliin lisätty "⇔" pois, ja klikkaa oikealle-nappia vielä kerran! Jäljelle jäi vain jälkimmäinen päättely, joka itsessään on oikein.

Jos `assume ¬F ∧ T;` pyyhitään pois, niin MathCheck sallii F:n saada arvoksi myös T ja T:n saada F, ja katsoo päättelyt virheellisiksi. Mutta kun päättelyissä olevat `F` :t ja `T` :t muutetaan muotoon `FF` ja `TT` , niin päättelyt kelpaavat jälleen. Kokeile jommallakummalla päättelyllä! (Pois pyyhityt oletukset palaavat takaisin lataamalla tehtävä uudelleen.)

Sitten kokeile, mitä tapahtuu, jos pyyhit pois myös `assume ¬H;` ja klikkaat oikealle-nappia.

Olemme nyt perin juurin ruotineet kaavoja $\neg(A \wedge H)$ ja $\neg A \wedge H$ tapauksissa, joissa tuuli ei humise hiljaa. Onkohan niillä eroa tapauksissa, joissa tuuli humisee hiljaa? Sen voi kysyä MathCheckiltä! Kirjoita tai (maalaa, kopioi ja pudota) äskeisen tehtävän oletuslaatikkoon `assume H;` ja vastauslaatikkoon $\neg(A \wedge H) \Leftrightarrow \neg A \wedge H$. Sitten klikkaa oikealle-nappia!

Myös $P \wedge Q \Leftrightarrow Q \wedge P$ eli $kaava1 \wedge kaava2 \Leftrightarrow kaava2 \wedge kaava1$ on logiikan laki. Se sanoo, että "ja" on osakaavojensa järjestyksestä riippumaton. Esimerkiksi "aurinko paistaa ja tuuli humisee hiljaa" tarkoittaa samaa kuin "tuuli humisee hiljaa ja aurinko paistaa". Tämä on samankaltainen asia kuin se, että lukujen yhteenlasku on yhteenlaskettavien järjestyksestä riippumaton. Esimerkiksi $2+3=3+2$.

Kaikki laskutoimitukset eivät ole järjestyksestä riippumattomia. Esimerkiksi vähennyslasku ei ole, sillä $5-2=3$ mutta $2-5=-3$. Sivulla **??** tulemme näkemään, että logiikan "→" ei ole järjestyksestä riippumaton.

Vähennyslaskussa sulkeiden paikka on olennainen: esimerkiksi $(9-5)-3=4-3=1$, mutta $9-(5-3)=9-2=7$. Yhteenlasku ei ole herkkä sulkeiden paikalle, sillä kaikille luvuille x, y ja z pätee $(x+y)+z=x+(y+z)$. Esimerkiksi $(9+5)+3=14+3=17$ ja $9+(5+3)=9+8=17$. "Ja" on tässä suhteessa yhteenlaskun kaltainen. Toisin sanoen: kaikille kaavoille P, Q ja R pätee $(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$.

Myös "∨" on näillä tavoilla laskujärjestyksestä riippumaton. Siis $P \vee Q \Leftrightarrow Q \vee P$ ja $(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$, kun P:n tilalle laitetaan mikä tahansa kaava, ja samoin Q:n ja R:n tilalle.

Toisaalta, kuten sivulla 13 totesimme, ”hän ottaa (teetä tai kahvia) ja maitoa” ei tarkoita samaa kuin ”hän ottaa teetä tai (kahvia ja maitoa)”. Siispä $(P \vee Q) \wedge R$ ei tarkoita samaa kuin $P \vee (Q \wedge R)$. Huomaamme, että laskujärjestys voi muuttua tärkeäksi, jos kaavassa esiintyy sekä ”$\wedge$” että ”$\vee$”.

Kaksi erityisen kiinnostavaa $\Leftrightarrow$-muotoista lakia on nimetty Augustus De Morganin (1806–1871) mukaan. Tutustumme toiseen niistä vertaamalla väitteitä ”hän ei ota sekä maitoa että sokeria” ja ”hän ei ota maitoa tai hän ei ota sokeria”. Täytä täältä löytyvät kaksi taulukkoa kuvan 2 tapaisesti.

Maidon ottaminen tai ottamatta jättäminen yhdistettynä sokerin ottamiseen tai ottamatta jättämiseen muodostaa kaikkiaan neljä eri tapausta. Kummassakin taulukossa on rivi niistä jokaiselle.

Ensimmäinen taulukoista voidaan täyttää seuraavasti. Jos hän ei ota maitoa, niin hän ei ota sekä maitoa että sokeria. Niinpä kaksi ylintä riviä kuuluu valita. Jos hän ottaa maitoa mutta ei sokeria, niin silloinkaan hän ei ota sekä maitoa että sokeria. Siis kolmaskin rivi kuuluu valita. Neljännellä rivillä hän ottaa sekä maitoa että sokeria. Se on suoraan vastoin väitettä, joten neljättä riviä ei kuulu valita.

Jälkimmäinen taulukoista voidaan täyttää seuraavasti. Kahdella ensimmäisellä rivillä toteutuu ”ei maitoa”, joten ne kuuluu valita. Kolmannella rivillä toteutuu ”ei sokeria”, joten sekin kuuluu valita. Neljännellä rivillä kumpikaan ei toteudu, joten sitä ei kuulu valita.

Taulukoihin tuli siis rastit samoille riveille: kumpaankin tuli rasti kolmelle ensimmäiselle riville, eikä tullut rastia neljännelle riville. Kummassakin taulukossa oli rivi jokaiselle mahdolliselle tapaukselle. Nämä yhdessä osoittavat, että ”ei sekä maitoa että sokeria” ja ”ei maitoa tai ei sokeria” ovat yhtäaikaa totta ja yhtäaikaa epätotta. Toisin sanoen, $\neg(M \wedge S) \Leftrightarrow \neg M \vee \neg S$.

Tälläkään kertaa olennaista ei ollut, mitä väitteitä kirjaimet edustivat, vaan olennaista oli vain sanojen ”ja”, ”tai” ja ”ei” vuorovaikutus. Niinpä saatu yhtäpitävyys pätee, vaikka M:n ja S:n tilalle laitettaisiin mitkä tahansa kaavat. Olemme päätelleet seuraavan lain: $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ eli $\neg(kaava1 \wedge kaava2) \Leftrightarrow \neg\, kaava1 \vee \neg\, kaava2$. Se on toinen De Morganin laeista.

Harjoittele De Morganin lain käyttöä jo tutuksi tulleella esimerkillä täällä!

Todistimme lain $\neg(M \wedge S) \Leftrightarrow \neg M \vee \neg S$ kokeilemalla kaikki totuusarvojen yhdistelmät, jotka siinä esiintyvät muuttujat (tässä tapauksessa M ja S) voivat saada. Kaikkien mahdollisuuksien kokeileminen toimii hyvin silloin kun muuttujia on vähän ja kukin niistä voi saada vain muutaman eri arvon. Ikävä kyllä sen työmäärä kasvaa rajusti muuttujien ja niiden mahdollisten arvojen määrän kasvaessa. Se kasvaa tietokoneillekin liian suureksi muutamalla kymmenellä tai muutamalla sadalla muuttujalla. Lisäksi ihmisten tekemänä se on virhealtista.

Siksi tarvitaan muitakin keinoja tutkia tai todistaa lakeja. Sivuilla 18 ja 19 tutkimme yhtäpitävyyttä $\neg(A \wedge H) \Leftrightarrow \neg A \wedge H$ sijoittamalla H:hon ensin F ja sitten T. Ensimmäisellä sijoituksella se sieveni selvästi epäpätevään muotoon $\mathsf{T} \Leftrightarrow \mathsf{F}$. Niinpä $\neg(A \wedge H) \Leftrightarrow \neg A \wedge H$ ei ole laki. Toisella sijoituksella se sieveni pätevään muotoon $\neg A \Leftrightarrow \neg A$. Se ei riitä kumoamaan sitä, että ensimmäisellä sijoituksella tuli epäpätevä lopputulos, mutta riittää osoittamaan, että $\neg(A \wedge H)$ ja $\neg A \wedge H$ eivät eroa toisistaan silloin, kun H on tosi.

Totuusarvoja saavien muuttujien tapauksessa on usein todella kätevää valita jokin muuttuja, sijoittaa siihen vuoron perään F ja T, sieventää "$\Leftrightarrow$":n molemmat puolet ja verrata lopputuloksia. Edellä mainittujen lakien $kaava \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$, $kaava \wedge \mathsf{T} \Leftrightarrow kaava$ ja niin edelleen avulla puolet sievenevät tyypillisesti paljon vähällä vaivalla. Usein lopputuloksista näkee heti, ovatko ne totta yhtäaikaa ja vain yhtäaikaa. Jollei näe, niin voidaan valita toinenkin muuttuja ja sijoittaa siihen vuoron perään F ja T, tai voidaan käyttää jotain muuta keinoa kuten kaikkien mahdollisuuksien kokeileminen.

Kokeile F:n ja T:n sijoittamista lakiin $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ täällä!

Kokeile F:n ja T:n sijoittamista yhtäpitävyyteen $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$ täällä!

Kumpikaan edellisistä keinoista ei toimi ollenkaan kokonaisluku- eikä reaalilukumuuttujille, sillä ne voivat saada loputtomasti eri arvoja. Seuraavassa esimerkissä havainnollistetaan muutamia tärkeitä päättelykeinoja, jotka toimivat hyvin monenlaisista asioista pääteltäessä, kuten luvuista.

Aloitamme äsken saadusta laista $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$. Yhtäpitävyys on symmetrinen ominaisuus: aina kun $kaava1$ ja $kaava2$ ovat joko molemmat totta tai kumpikaan ei ole totta, niin myös $kaava2$ ja $kaava1$ ovat joko molemmat totta tai kumpikaan ei ole totta. Toisin sanoen, jos $kaava1 \Leftrightarrow kaava2$, niin $kaava2 \Leftrightarrow kaava1$. Kerro täällä, minkälaiseksi $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ muuttuu symmetriaa soveltamalla!

Koska juuri saatu yhtäpitävyys on laki, se pätee kun P:n tilalle laitetaan mikä tahansa kaava ja Q:n tilalle laitetaan mikä tahansa kaava (tarvittaessa sulkeet lisäten). Nyt laitamme P:n tilalle $\neg P$ ja Q:n tilalle $\neg Q$. Mitä siten saadaan? Tarvittavat vastauslaatikot ovat edellisten vastauslaatikoiden alapuolella.

Sitten sievennämme edellistä tulosta lakia $\neg\neg kaava \Leftrightarrow kaava$ hyödyntämällä niin paljon kuin mahdollista. Kuten sivulla 18 kerrottiin, tämän luvun kaavoille pätee, että jos $kaava1 \Leftrightarrow kaava2$, niin $isompi\_kaava(kaava1) \Leftrightarrow isompi\_kaava(kaava2)$. Siksi on sallittua sieventää kaavaa tekemällä sievennyksiä sen osakaavoihin. Vastauslaatikot ovat kolmantena edellä mainitusta linkistä alkavassa ryhmässä.

Neljännessä vaiheessa käytämme uudelleen lupaa sieventää kaavan osaa. Tällä kertaa sieventämisessä käytettävä laki on edellinen tulos, ja $isompi\_kaava(X)$ on

$\neg X$.

Lienee helppo arvata, mitä viimeisessä vaiheessa pitää tehdä. Tee se!

Kaiken kaikkiaan olemme päätelleet seuraavasti:

$\neg(P \wedge Q)$	$\Leftrightarrow$	$\neg P \vee \neg Q$	aiemmin perusteltu De Morganin laki
$\neg P \vee \neg Q$	$\Leftrightarrow$	$\neg(P \wedge Q)$	$\Leftrightarrow$:n symmetrisyys
$\neg\neg P \vee \neg\neg Q$	$\Leftrightarrow$	$\neg(\neg P \wedge \neg Q)$	P:n ja Q:n tilalle $\neg P$ ja $\neg Q$
$P \vee Q$	$\Leftrightarrow$	$\neg(\neg P \wedge \neg Q)$	$\neg\neg$ *kaava* $\Leftrightarrow$ *kaava*
$\neg(P \vee Q)$	$\Leftrightarrow$	$\neg\neg(\neg P \wedge \neg Q)$	edellinen sovellettuna kaavaan $\neg X$
$\neg(P \vee Q)$	$\Leftrightarrow$	$\neg P \wedge \neg Q$	$\neg\neg$ *kaava* $\Leftrightarrow$ *kaava*

Lopputulos voidaan ilmaista niin, että P:n ja Q:n tilalla ovat *kaava*1 ja *kaava*2:

$$\neg(\mathit{kaava}1 \vee \mathit{kaava}2) \;\Leftrightarrow\; \neg\,\mathit{kaava}1 \wedge \neg\,\mathit{kaava}2$$

Lopputulos on muuten sama kuin lähtökohtana käytetty De Morganin laki, mutta $\wedge$ ja $\vee$ ovat vaihtaneet keskenään paikkoja. Sekin on De Morganin laki. Olisi ollut kätevää sanoa, että $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ on ensimmäinen De Morganin laki ja $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$ on toinen De Morganin laki, mutta se olisi ollut hieman väärin, koska ei ole olemassa sopimusta siitä, kumpi niistä on ensimmäinen ja kumpi toinen.

De Morganin lait paljastavat, että logiikassa vallitsee hyvin pitkälle duaalisuus: jos kaavassa vaihdetaan jokainen T ja F keskenään, vaihdetaan jokainen $\wedge$ ja $\vee$ keskenään (tarvittaessa sulkeita lisäten) ja jokaisen totuusarvoisen muuttujan eteen lisätään tai edestä poistetaan $\neg$, ja jos siinä ei esiinny mainittujen symboleiden lisäksi muita symboleita, niin lopputulos on yhtäpitävä alkuperäisen kanssa. Tämä duaalisuus tulee säilymään kun logiikan kieltä laajennetaan luvussa 3. Loppumattomiin se ei kuitenkaan säily, vaan tulemme näkemään sen menevän rikki luvussa 7.

De Morganin lait ovat loppujen lopuksi hyvin arkijärkisiä. Niinpä, vaikka ne on nimetty 1800-luvulla eläneen henkilön mukaan, ne tunnettiin jo antiikin aikana ja niistä kirjoitettiin myös keskiajalla. Kuten Wikipedian sivu ”De Morgan’s Laws” toteaa (lyhentäen suomennettuna): ”De Morganin lait on helppo todistaa, ja ne voivat jopa näyttää itsestäänselvyyksiltä. Yhtä kaikki, ne ovat hyödyllisiä pätevässä päättelyssä.”

2.4 Hieman tietokoneiden perusosista

3 Predikaattilogiigan kaavat

Tässä luvussa jatketaan edellisessä luvussa aloitettua pohdintaa siitä, miten logiikan kielellä esitetään väitteitä. Pääpaino on niin sanotussa ensimmäisen kertaluvun logiikassa, mutta myös korkeammista kertaluvuista kerrotaan sen verran, että niiden ero ensimmäiseen kertalukuun käy ilmi. Myös esitellään predikaattilogiikan lakeja ja jatketaan päättelemiseen tutustumista. Lisäksi kerrotaan, mitä nykyaikaisessa logiikassa tarkoittaa, että väite on tosi.

3.1 Olemassaolokvanttori

Tähänastisilla keinoilla voidaan kirjoittaa kaava, joka on aina tosi jos puheenaiheena on kokonaisluvut, mutta ei ole aina tosi jos puheenaiheena on reaaliluvut. Esimerkiksi $x \leq 7 \vee x \geq 8$ on sellainen. Kokeile sitä täällä!

Sitten keksi kaava, joka on aina tosi jos puheenaiheena on luonnolliset luvut, mutta ei ole aina tosi jos puheenaiheena on kokonaisluvut. Kirjoita se hieman edellistä alempana olevaan vastauslaatikkoon! (Kaavassa saa olla vain sellaisia symboleita, jotka ovat MathCheckissä käytössä sekä luonnollisilla luvuilla että kokonaisluvuilla. Muun muassa ”$-$” ei käy, koska esimerkiksi $3-5$ ei ole määritelty luonnollisilla luvuilla.)

Jossain määrin vaikeampaa on keksiä kaava, joka on aina tosi jos puheenaiheena on reaaliluvut, mutta ei ole aina tosi jos puheenaiheena on kokonaisluvut. Se ei onnistu lisäämällä äsken käytetyn kaavan $x \leq 7 \vee x \geq 8$ ympärille sulkeet ja eteen ”$\neg$”. Mieti ensin miksei, ja sitten kokeile kolmannessa vastauslaatikossa!

Kaava $x \leq 7 \vee x \geq 8$ sanoo, että x ei ole seitsemän ja kahdeksan välillä. Koska x:stä ei ole tehty mitään oletuksia, voi x olla mikä luku tahansa. Niinpä kaava väittää, että mikään luku ei ole seitsemän ja kahdeksan välillä. Se todellakin on tosi kokonaisluvuilla, koska mikään kokonaisluku ei ole sillä välillä. Se ei ole tosi reaaliluvuilla, koska $7\frac{1}{2}$ on reaaliluku ja on sillä välillä.

Kun äskeisen kaavan ympärille lisätään sulkeet ja eteen ”$\neg$”, saadaan kaava, joka sanoo, että x on seitsemän ja kahdeksan välillä. Nytkin x voi olla mikä luku tahansa. Siksi muokattu kaava sanoo, että jokainen luku on seitsemän ja kahdeksan välillä. Se ei ole tosi sen koommin kokonais- kuin reaaliluvuilla, koska esimerkiksi 0 on kokonaisluku ja reaaliluku, joka ei ole sillä välillä.

Saisimme halutunlaisen kaavan aikaiseksi, jos saisimme sanan ”jokainen” muutettua sanaksi ”jokin”. Haluamme siis kaavan, joka sanoo, että jokin luku on seitsemän ja kahdeksan välillä. Se on totta reaaliluvuilla, koska muun muassa $7\frac{1}{2}$ on reaaliluku ja on sillä välillä, mutta ei ole totta kokonaisluvuilla, koska 7:n ja 8:n välissä ei ole yhtään kokonaislukua. Saamme tällaisen kaavan lisäämällä äskeisen kaavan eteen `EE x:` . Kokeile! Jos kaikki sujui kuten piti, niin MathCheckin palaute alkaa $\exists x : \neg(x \leq 7 \vee x \geq 8) \equiv \mathsf{T}$, missä ”$\equiv$” on reaaliluvuilla vihreällä ja kokonaisluvuilla punaisella.

Symboli ”$\exists$” on nimeltään *olemassaolokvanttori* (englanniksi *existential quantifier*). Sen kirjoitustapa vaihtelee. Teoreettisessa kirjallisuudessa on tapana jättää kaksoispiste pois ja käyttää enemmän sulkeita, tyyliin $\exists x(\neg(x \leq 7 \vee x \geq 8))$ tai $(\exists x)(\neg(x \leq 7 \vee x \geq 8))$. Jotkut käyttävät kaksoispisteen tilalla pistettä tyyliin $\exists x.\neg(x \leq 7 \vee x \geq 8)$. Tällä kurssilla käytettävä tapa tavoittelee sitä, että kaavan rakenne olisi helppo hahmottaa varsinkin ottaen huomioon, että tulemme sivulla 48 ottamaan käyttöön myös merkinnän $\exists x$; *rajoite* : *kaava*.

Kirjoita seuraavaan vastauslaatikkoon kaava, joka on aina tosi kokonaisluvuilla mutta ei ole aina tosi luonnollisilla luvuilla!

Oletko koskaan tullut miettineeksi, mistä johtuu, että heinäkuun ensimmäinen on aina sama viikonpäivä kuin huhtikuun ensimmäinen? Myös heinäkuun toinen on aina sama viikonpäivä kuin huhtikuun toinen ja niin edelleen aina 30. päivään saakka (ja vappu on aina sama viikonpäivä kuin heinäkuun viimeinen). Se johtuu siitä, että viikossa on seitsemän päivää, ja huhti-, touko- ja kesäkuussa on yhteensä seitsemällä jaollinen määrä päiviä. Huhtikuussa on 30, toukokuussa 31 ja kesäkuussa 30 päivää, eli yhteensä 91 päivää, ja $91 = 7 \cdot 13$.

Nyt ei keskitytä lukujen jaollisuusoppiin, mutta pari asiaa on tarpeen tietää. Jos kolme desilitraa mehua pitää jakaa kahdelle lapselle, niin kummallekin voi antaa puolitoista desilitraa. Mutta kolmen ilmapallon jakaminen kahdelle lapselle ei suju yhtä hienosti, sillä jos ilmapallon puolittaa, niin se menee täysin rikki. Tämäntapaisista syistä on tavallisen jakolaskun lisäksi olemassa myös kokonaislukujen jakolasku, joka lopettaa jakamisen siinä vaiheessa, kun jouduttaisiin pilkkomaan jaettavia esineitä osiksi. Jakamatta jäävää määrää kutsutaan *jakojäännökseksi* (*remainder*). Kunkin saajan saamaa määrää kutsutaan tavalliseen tapaan *osamääräksi* (*quotient*).

Kokonaislukujen jakolaskun osamäärälle ja jakojäännökselle on käytössä monia merkintöjä. MathCheck käyttää merkintöjä ”div” ja ”mod”. Niitä ei tulla tarvitsemaan seuraavissa tehtävissä — itse asiassa niiden käyttö on kielletty, koska tarkoitus ei ole opetella niiden vaan olemassaolokvanttorin ominaisuuksia. Mutta MathCheckin tuottamassa palautteessa käytetään merkintää ”mod”, joten on hyvä tietää, että se tarkoittaa jakojäännöstä.

Puheenaiheena on kokonaisluvut. Kirjoita tänne kaava, joka sanoo että n on

seitsemällä jaollinen!

Edellä piti kirjoittaa kaava, joka sanoo, että jokin luku kerrottuna seitsemällä tuottaa luvun n. Jos kirjoitit sen aiempien mallien mukaisesti, niin siinä esiintyy muuttuja x. Se edustaa sitä ”jotakin lukua”, joka kerrottuna seitsemällä tuottaa luvun n. Ei ole olennaista, että ”jotakin lukua” edustavan muuttujan nimi on nimenomaan x. Olennaista on vain, että eri asioita tarkoittavat muuttujat eivät sekoitu toisiinsa. Tässä tapauksessa se tarkoittaa, että x:n tilalla voi olla mikä tahansa muuttujan nimi paitsi n.

Jos x:n tilalle laitetaan n, niin saadaan $\exists n : n = 7n$. Kopioi se viimeisimmän vastauksesi tilalle ja kokeile, mitä MathCheck antaa palautteeksi. Palautteen pitäisi näyttää suunnilleen tältä:

> $n \bmod 7 = 0 \;\Leftrightarrow\; \exists n : n = 7n$
>
> Relation does not hold when $n = -1$
>
> left $\equiv$ F
>
> right $\equiv$ T

Yritä hetken aikaa ymmärtää palaute ja vastata tehtäväryhmän muihin tehtäviin. Sitten jatka lukemista.

Palaute sanoo, että jos $n = -1$, niin mallivastaus $n \bmod 7 = 0$ ei ole tosi mutta vastaukseksi kopioitu kaava $\exists n : n = 7n$ on tosi. Palautteen ymmärtämiseksi on tärkeää huomata, että n kaavassa $\exists n : n = 7n$ on eri muuttuja kuin n kaavassa $n \bmod 7 = 0$, vaikka niillä on sama nimi. MathCheck yrittää auttaa tämän huomaamista näyttämällä ensin mainitun eri värillä. Kaavan $\exists n : n = 7n$ muuttujan n nimellä ei ole väliä. Kaavan merkitys ei muutu, vaikka sen tilalle vaihdettaisiin jokin muu nimi. Esimerkiksi $\exists x : x = 7x$ ja $\exists a : a = 7a$ tarkoittavat samaa kuin $\exists n : n = 7n$.

Kaava $\exists n : n = 7n$ sanoo, että on olemassa sellainen luku, että kun sen kertoo seitsemällä, saadaan tämä luku itse. Se on totta, sillä nolla on sellainen luku. Kaava $\exists n : n = 7n$ ei puhu siitä n, joka esiintyy kaavassa $n \bmod 7 = 0$ ja josta MathCheck puhuu palautteessaan. Tämä käy selvemmäksi, kun palaute esitetään sanallisesti:

> n on jaollinen seitsemällä $\Leftrightarrow$ on olemassa sellainen luku, että kun sen kertoo seitsemällä, saadaan se itse
>
> Tämä ei päde kun $n = -1$. Silloin vasen puoli on epätosi ja oikea puoli on tosi.

Oikea puoli on tosi riippumatta palautteessa mainitun n:n arvosta, koska se ei todellisuudessa puhu palautteen n:stä. Sen sijaan vasen puoli riippuu siitä. Palautteessa ei voi esiintyä $n = 0$, koska myös vasen puoli olisi sillä tosi, joten se ei olisi

vastaesimerkki. Palautteessa voisi esiintyä $n = 1$ tai $n = 2$ tai $n = 6$ tai $n = 100$, koska vasen puoli on epätosi jokaisella niistä. On äärettömän monta eri n:n arvoa, jotka voisivat esiintyä palautteessa. Sivulla 59 selviää, miksi MathCheck valitsi niiden joukosta arvon -1.

Kokeile, miten MathCheckin palaute muuttuu, kun vaihdat vastauksessa $\exists n : n = 7n$ jokaisen n:n tilalle a:n. MathCheck näyttää saman vastaesimerkin kuin aiemmin, mutta nyt MathCheck ei näytä jälkimmäistä muuttujaa eri värillä, koska se ei ole samanniminen kuin ensimmäinen muuttuja.

MathCheck ei anna palautteessaan arvoa a:lle seuraavasta syystä. Palautteessaan MathCheck antaa arvot siihenastisen päättelyketjun vapaille muuttujille mutta ei sidotuille. Se riittää kertomaan tilanteen, jossa kaava ei ole tosi tai päättelyaskel ei ole pätevä. Sidotun muuttujan arvosta puhuminen palautteessa ei olisi aina edes mielekästä. Esimerkiksi kokonaisluvuilla $\exists x : x^2 = 2$ ei ole tosi. Se on epätosi siksi, että sellaista kokonaislukua x ei ole olemassa, jonka neliö on 2. Sitä, että sopivaa lukua ei ole olemassa, ei voi perustella antamalla palautteessa jokin luku.

Seuraavaksi tutkimme olemassaolokvanttorin käyttäytymistä muutamalla tehtävällä. Niistä kolmannen kohdalla vastaan tulee tärkeä ilmiö. Onnea matkaan!

Kirjallisuudessa on erilaisia käytäntöjä sen suhteen, kuinka pitkälle kvanttorin vaikutusalue ulottuu. Kyse on esimerkiksi siitä, tarkoittaako $\exists x : n = 7x \wedge n > 4$ samaa kuin $(\exists x : n = 7x) \wedge n > 4$ vai samaa kuin $\exists x : (n = 7x \wedge n > 4)$. MathCheck noudattaa käytäntöä, jonka mukaan kvanttorin vaikutusalue loppuu vasta kun koko kaava loppuu tai vastaan tulee ”)”, jota vastaava ”(” oli ennen kvanttoria. Niinpä MathCheckille $\exists x : n = 7x \wedge n > 4$ tarkoittaa samaa kuin $\exists x : (n = 7x \wedge n > 4)$. Tämä käytäntö valittiin, koska se tyypillisesti vähentää sulkeiden tarvetta.

Symbolit ”$\Leftrightarrow$”, ”$\Rightarrow$”, ”$\Leftarrow$” ja ”$\equiv$” eivät esiinny kaavojen sisällä vaan kaavojen välissä, joten kvanttorin vaikutusalue loppuu myös niihin.

Kenties vastasit $(\neg\exists x : n = 7x) \wedge n > 4$ mutta MathCheck näyttikin palautteessaan $\neg(\exists x : n = 7x) \wedge n > 4$. Ne tarkoittavat samaa, koska ”$\neg$” lasketaan ennen kuin ”$\wedge$”. Jommatkummat sulkeet on oltava, koska muutoin myös $n > 4$ joutuisi $\neg$:n piiriin. MathCheckin käyttämässä sisäisessä esitysmuodossa ei ole tietoa, kummassa kohdassa sulkeet alun perin olivat. Siksi MathCheck saattaa tulostaa ne eri kohtaan kuin missä ne olivat vastauksessa.

Jokainen puheenaiheen arvon sisältävä muuttuja on joko *vapaa* (*free*) tai *sidottu* (*bound*). Muuttuja on sidottu jos ja vain jos se on luotu kvanttorilla. Vapaata ja sidottua voi havainnollistaa esimerkiksi lauseella ”hän on ystävällinen kaikille”. Se on tosi tai epätosi sen mukaan, ketä ”hän” tarkoittaa. ”Hän” viittaa yksittäiseen ihmiseen, mutta riippuu asiayhteydestä, kuka tämä ihminen on. Vapaa muuttuja käyttäytyy juuri näin. Sen sijaan ”kaikille” ei viittaa keneenkään yksittäiseen ihmiseen, vaan jokaiseen ihmiseen. Siksi sen merkitys ei riipu asiayhteydestä (paitsi

siltä osin, että puhutaanko kaikista ihmisistä vai esimerkiksi kaikista kokonaisluvuista). Sidottu muuttuja käyttäytyy juuri niin.

”Kaikille” vastaa niin sanottua kaikkikvanttoria ”$\forall$”. Siihen keskitytään luvussa 3.3, mutta jo nyt voi toki kertoa, että $\forall x : kaava$ on tosi jos ja vain jos valitaanpa x:lle mikä tahansa arvo, niin *kaava* on tosi. Vastaavasti $\exists x : kaava$ on tosi jos ja vain jos on olemassa ainakin yksi sellainen arvo, että kun x saa sen, niin *kaava* on tosi. ”Hän on ystävällinen kaikille” tarkoittaa samaa kuin ”hän ei ole epäystävällinen kenellekään”. Siinä ”kenellekään” vastaa olemassaolokvanttoria ”$\exists$”. Jos ”a on epäystävällinen b:lle” ilmaistaan $a \ddot{\sim} b$, ja jos h edustaa ”häntä”, niin ”hän ei ole epäystävällinen kenellekään” voidaan ilmaista kaavana $\neg\exists x : h \ddot{\sim} x$. Sama voidaan ilmaista myös kaavana $\forall x : \neg(h \ddot{\sim} x)$. Kummassakin h on vapaa ja x on sidottu.

Kaava on *avoin* (*open*), jos ja vain jos siinä esiintyy ainakin yksi vapaa muuttuja. Päinvastaisessa tapauksessa kaava on *suljettu* (*closed*). Esimerkiksi ”hän on ystävällinen kaikille” on avoin, koska ”hän” on vapaa muuttuja. Sen sijaan ”joku on ystävällinen kaikille” on suljettu, koska ”joku” vastaa $\exists$:lla ja ”kaikille” vastaa $\forall$:lla luotua muuttujaa, joten ne vastaavat sidottuja muuttujia. Kaava $\exists x : n = 7x$ on avoin, koska n on siinä vapaa, mutta $\exists n : n = 7n$ on suljettu, koska n on siinä sidottu eikä siinä esiinny muita muuttujia. Kaava $n > 4 \wedge \exists n : n = 7n$ on avoin. Siinä esiintyy kaksi eri muuttujaa, joiden kummankin nimi on n. Toinen niistä on vapaa ja toinen sidottu. Sama pätee kaavalle $(\exists n : n = 7n) \wedge n > 4$. Mutta $\exists n : n = 7n \wedge n > 4$ on suljettu, sillä sen jokainen n on $\exists$:n alaisuudessa.

Suljettu kaava on joko tilanteesta riippumatta tosi tai tilanteesta riippumatta epätosi. Avoimen kaavan totuusarvo voi riippua tilanteesta, eli siitä, mitkä arvot vapaille muuttujille annetaan. Esimerkiksi kokonaisluvuilla $\exists x : n = 7x$ on tosi kun $n = 7$, mutta epätosi kun $n = 8$. Kuitenkin myös avoin kaava voi olla tilanteesta riippumatta tosi tai tilanteesta riippumatta epätosi. Esimerkiksi kokonaisluvuilla $\exists x : n < x$ on n:n arvosta riippumatta tosi, sillä olipa n mikä tahansa kokonaisluku, on esimerkiksi $n + 1$ suurempi kuin n. Ihmisistä puhuttaessa ”hän pystyy elämään kuukausia ilman vettä” on avoin, ja se on epätosi riippumatta siitä, kuka ”hän” on.

Sekä *sentence* että *closed formula* tarkoittavat suljettua kaavaa.

Täällä on tehtäviä vapaiden ja sidottujen muuttujien sekä avoimien ja suljettujen kaavojen harjoittelemiseksi.

Sidottu muuttuja lakkaa olemasta, kun sen kvanttorin vaikutusalue loppuu, jolla muuttuja luotiin. Siksi esimerkiksi kaavassa $(\exists i : n = 7i) \wedge 1 \leq i \leq n \wedge \exists i : n = 4i$ esiintyy kaikkiaan neljä muuttujaa, joista kolmen nimi on i. Vapaa muuttuja on olemassa koko kaavan ajan. Siihen ei kuitenkaan pääse käsiksi niissä osissa kaavaa, joissa on olemassa toinen samanniminen muuttuja, koska siellä muuttujan nimi viittaa jälkimmäiseen. Esimerkiksi edellä osuudessa $\exists i : n = 4i$ ei päästä käsiksi vapaaseen i:hin, koska siellä i viittaa jälkimmäisellä $\exists$:lla luotuun i:hin. Myös on

mahdollista, että sidottuun muuttujaan ei pääse käsiksi, koska samassa kohdassa on olemassa toinenkin samanniminen sidottu muuttuja.

MathCheck näyttää muuttujan esiintymän tavallisesta poikkeavalla värillä jos ja vain jos samassa kohdassa on olemassa toinenkin samanniminen muuttuja. MathCheck siis varoittaa, että tässä kohdassa nimi saattaa tarkoittaa eri muuttujaa kuin haluttiin. Poikkeava väri on valittu siten, että se erottuu punaisesta hyvin mutta mustasta huonommin. Näin siksi, että varoitus nimien mahdollisesta sekaantumisesta saattaa auttaa virheellisen kaavan korjaamisessa, mutta asialla ei ole suurta väliä kun kaava on oikein. Toinen muuttuja voi olla peräisin saman kaavan lisäksi myös aikaisemmasta kaavasta. Sen ansiosta varoitus muuttujien mahdollisesta sekoittumisesta tulee myös silloin, kun sidottu muuttuja on samanniminen kuin tehtävänasettelussa esiintyvä muuttuja.

Sidotun muuttujan nimen saa valita vapaasti, kunhan se ei sekaannu minkään muun muuttujan nimeen. Jos sidotulle muuttujalle valitaan nimi, jota ei käytetä mihinkään muuhun, niin sekaannusta ei voi syntyä. Valitettavasti se saattaa johtaa pitkien kaavojen tapauksessa epäluonteviin nimivalintoihin. Käytetyn nimen käyttäminen uudelleen on turvallista, jos nimi ei esiinny vapaana missään eikä sidottuna puheena olevan kvanttorin kohdalla. Esimerkiksi jos i ei esiinny vapaana, niin $(\exists i : \ldots) \wedge (\exists i : \ldots)$ on turvallinen, koska ensimmäinen i lakkaa olemasta ennen kuin jälkimmäinen i luodaan, mutta $(\exists i : \ldots \wedge (\exists i : \ldots))$ saattaa olla vaarallinen, koska jälkimmäinen i luodaan kohdassa, jossa edellinen i on yhä olemasssa.

Jatka olemassaolokvanttorin käyttäytymiseen tutustumista ja MathCheckin antaman palautteen hyödyntämisen harjoittelemista täällä!

Riippuen siitä miten korjasit vastauslaatikossa olleen kaavan, MathCheck saattaa näyttää palautteessa osan muuttujista tavallisesta poikkeavalla värillä. Se ei tarkoita että vastaus olisi väärin, vaan se on vain MathCheckin tapa varoittaa siitä, että eri värillä näytetty muuttuja saattaa kaavaa lukiessa sekoittua toiseen samannimiseen muuttujaan. On helppo muuttaa kaavaa siten, että sekoittumisen vaara poistuu. Tässä esimerkissä se onnistuu ainakin kahdella eri keinolla: voidaan lisätä sulkeet sopiviin kohtiin tai voidaan vaihtaa jonkin sidotun muuttujan nimi. (Sivulla 33 paljastuu, miksi vastaus kelpaa ilman sulkeita.)

Seuraavaksi käymme läpi joukon lakeja, joita ”∃” noudattaa. Johdantona aihepiiriin käsittelemme hiukan vinksahtaneen esimerkin. Otamme käyttöön sanat ”vinksis” ja ”töksis”. Ne ovat ihmisille mahdollisia ominaisuuksia, mutta tämän enempää niistä ei kerrota. Mitkä seuraavista ovat totta?

- Jos joku professori on vinksis tai töksis, niin joku professori on vinksis tai joku professori on töksis.
- Jos joku professori on vinksis tai joku professori on töksis, niin joku professori on vinksis tai töksis.

- Jos joku professori on vinksis ja töksis, niin joku professori on vinksis ja joku professori on töksis.
- Jos joku professori on vinksis ja joku professori on töksis, niin joku professori on vinksis ja töksis.

Ensimmäinen on helppo hyväksyä todeksi. Jos joku professori on vinksis tai töksis, niin silloinhan hän on vinksis tai hän on töksis. Jos hän on vinksis, niin joku professori (nimittäin ainakin hän) on vinksis. Jos hän on töksis, niin joku professori (nimittäin ainakin hän) on töksis.

Jos joku professori on vinksis, niin hän on vinksis tai töksis, joten joku professori on vinksis tai töksis. Jos joku professori on töksis, niin hän on vinksis tai töksis, joten joku professori on vinksis tai töksis. Siksi toinen pätee.

Jos joku professori on vinksis ja töksis, niin hän on vinksis, joten joku professori on vinksis. Hän on myös töksis, joten joku professori on töksis. Niinpä joku professori on vinksis ja joku professori on töksis. Siis myös kolmas pätee.

On mahdollista, että joku professori on vinksis vaan ei töksis ja joku toinen professori on töksis vaan ei vinksis, mutta kukaan professori ei ole sekä vinksis että töksis. Siksi neljäs ei päde.

Sitten siirrymme takaisin vähemmän vinksahtaneeseen ja töksähtäneeseen kielenkäyttöön. Merkinnät *kaava*, *kaava*1 ja niin edelleen edustavat mitä tahansa kaavoja. Toisinaan on tarpeen puhua muuttujan x kaikista vapaista esiintymistä kaavassa, esimerkiksi siksi että niiden tilalle tullaan kohta laittamaan jotakin muuta kuin x, esimerkiksi $1+2y$. Silloin käytämme ennen sijoitusta merkintää $kaava(x)$ tai jotakin sen kaltaista, ja sijoituksen jälkeen merkintää $kaava(1+2y)$ tai jotakin sen kaltaista. Merkintä $kaava(x)$ ei lupaa, että x esiintyy vapaana. Jos x ei esiinny vapaana, niin $kaava(x)$ ja $kaava(1+2y)$ ovat täysin sama.

Useimmat lakimme ovat muodoltaan yhtäpitävyyksiä, joissa esiintyy sama kaava tai samat kaavat molemmilla puolilla. Sellainen ehdokas laiksi on epäpätevä, jos ja vain jos on olemassa yksikin tapa valita siinä esiintyvät kaavat ja niissä esiintyvien vapaiden muuttujien arvot siten, että toinen puoli on tosi ja toinen puoli ei ole tosi. Toisin sanoen, jos lakiehdokkaalle on yksikin vastaesimerkki, niin ehdokas ei ole laki; ja jos sille ei ole yhtään vastaesimerkkiä, niin se on laki.

Tämän luvun tähänastisissa tehtävissä on ollut muutamia, joissa kahdesta kaavasta on osoittautunut, että ne eivät ole yhtäpitävät. Niiden perusteella voidaan todeta muutama lakiehdokas epäpäteväksi. Täällä on kuusi lakiehdokasta, joista osa on päteviä ja loput eivät ole. Yritä valita epäpätevät! Jos et onnistu, niin se ei haittaa. Osoitamme myöhemmin loput päteviksi. Silloin epäpätevät paljastuvat sitä kautta, että niitä ei osoiteta päteviksi. Huomaa, että saattaa olla pienen lisäehdon puuttumisesta kiinni, että lakiehdokas ei ole pätevä.

Aloitamme lakien päteväksi osoittamisen laista $\exists x : kaava1(x) \vee kaava2(x) \Leftrightarrow \big(\exists x : kaava1(x)\big) \vee \exists x : kaava2(x)$. Merkintä "$(x)$" ei ole tässä välttämätön, mutta kirjoitamme sen korostamaan sitä, että x saattaa esiintyä vapaana $kaava1$:ssä ja/tai $kaava2$:ssa. Jotta emme vahingossa lukisi kaavan rakennetta väärin, käytämme ylimääräisiä sulkeita:

$$\exists x : \big(kaava1(x) \vee kaava2(x)\big) \;\Leftrightarrow\; \big(\exists x : kaava1(x)\big) \vee \big(\exists x : kaava2(x)\big)$$

Tarkastelemme muuttujia, jotka esiintyvät vapaina sen jommallakummalla tai molemmilla puolilla. Niihin ei kuulu x, koska sen kukin esiintymä on sidottu $\exists$:lla. Tarkastelemme mitä tahansa näiden muuttujien arvojen yhdistelmää.

Jos vasen puoli on siinä tosi, niin on olemassa sellainen x:n arvo, että kun muuttujien arvojen yhdistelmää täydennetään sillä, niin $kaava1(x) \vee kaava2(x)$ on tosi. Sillä yhdistelmällä $kaava1(x)$ tai $kaava2(x)$ tai molemmat on tosi. Jos $kaava1(x)$ on tosi, niin alkuperäisellä vapaiden muuttujien arvojen yhdistelmällä $\exists x : kaava1(x)$ on tosi. Muussa tapauksessa $kaava2(x)$ on tosi ja alkuperäisellä vapaiden muuttujien arvojen yhdistelmällä $\exists x : kaava2(x)$ on tosi. Molemmissa tapauksissa $\big(\exists x : kaava1(x)\big) \vee \big(\exists x : kaava2(x)\big)$ on alkuperäisellä yhdistelmällä tosi. Olemme osoittaneet, että jos lain vasen puoli on tosi, niin myös oikea on.

Jos oikea puoli on tosi alkuperäisellä vapaiden muuttujien arvojen yhdistelmällä, niin $\exists x : kaava1(x)$ tai $\exists x : kaava2(x)$ tai molemmat on siinä tosi. Jos $\exists x : kaava1(x)$ on tosi, niin muuttujien arvojen yhdistelmää voi täydentää sellaisella x:n arvolla, että $kaava1(x)$ on tosi. Muussa tapauksessa sama pätee $kaava2(x)$:lle. Molemmissa tapauksissa $kaava1(x) \vee kaava2(x)$ on täydennetyllä yhdistelmällä tosi. Siksi $\exists x : \big(kaava1(x) \vee kaava2(x)\big)$ on tosi alkuperäisellä vapaiden muuttujien arvojen yhdistelmällä. Olemme osoittaneet, että jos lain oikea puoli on tosi, niin myös vasen on. Laki on nyt todistettu.

Tämän todistuksen ymmärtämistä voi vahvistaa miettimällä, mitä tapahtuu, jos samaa yritetään siten, että symbolin "$\vee$" tilalla on "$\wedge$". Tarkastellaan taas mitä tahansa vapaiden muuttujien arvojen yhdistelmää. Jos $\exists x : \big(kaava1(x) \wedge kaava2(x)\big)$ on sillä tosi, niin on olemassa sellainen x:n arvo, että kun yhdistelmää täydennetään sillä, niin sekä $kaava1(x)$ että $kaava2(x)$ on tosi. Niinpä alkuperäisellä yhdistelmällä sekä $\exists x : kaava1(x)$ että $\exists x : kaava2(x)$ on tosi, joten myös $\big(\exists x : kaava1(x)\big) \wedge \big(\exists x : kaava2(x)\big)$ on tosi. Siis tässäkin tapauksessa aina kun vasen puoli on tosi, myös oikea puoli on tosi.

Mutta tässä tapauksessa päättely toiseen suuntaan ei mene läpi. Jos $\big(\exists x : kaava1(x)\big) \wedge \big(\exists x : kaava2(x)\big)$ on tosi alkuperäisellä vapaiden muuttujien arvojen yhdistelmällä, niin on olemassa x:n arvo, jolla täydentämällä $kaava1(x)$ on tosi; ja on olemassa x:n arvo, jolla täydentämällä $kaava2(x)$ on tosi. Mutta missään ei luvata, että ne ovat sama arvo. Siksi ei voida luvata, että $kaava1(x) \wedge kaava2(x)$ on tosi millään x:n arvolla. Niin ollen ei myöskään voida luvata, että $\exists x : \big(kaava1(x) \wedge kaava2(x)\big)$ on tosi.

Siis aina kun vasen puoli on tosi, myös oikea puoli on tosi, mutta ei luvata että sama pätisi toisinpäin. Tällaisen ilmaisemista varten on symboli "$\Rightarrow$". Olemme osoittaneet, että

$$\exists x : \big(kaava1(x) \wedge kaava2(x)\big) \;\Rightarrow\; \big(\exists x : kaava1(x)\big) \wedge \big(\exists x : kaava2(x)\big)$$

Todistuksemme ei ota kantaa siihen, onko vastakkainen suunta epäpätevä kaikilla kaavoilla, joillain mutta ei kaikilla kaavoilla, vai ei millään kaavoilla. Mutta on helppo keksiä esimerkkejä, jotka osoittavat keskimmäisen vaihtoehdon todeksi. Esimerkiksi $(\exists x : x = 1) \wedge (\exists x : x = 2)$ on tosi, mutta $\exists x : (x = 1 \wedge x = 2)$ ei ole. Toisaalta sekä $(\exists x : x = 1) \wedge (\exists x : x = 1)$ että $\exists x : (x = 1 \wedge x = 1)$ on tosi.

Myös lain $\exists x : \exists y : kaava(x,y) \Leftrightarrow \exists y : \exists x : kaava(x,y)$ voi osoittaa päteväksi tarkastelemalla siinä esiintyvien vapaiden muuttujien mitä tahansa arvojen yhdistelmää. Jos $\exists x : \exists y : kaava(x,y)$ on siinä tosi, niin sitä voi täydentää sellaisella x:n arvolla, että $\exists y : kaava(x,y)$ on täydennetyllä yhdistelmällä tosi. Se puolestaan tarkoittaa, että on olemassa sellainen y:n arvo, että $kaava(x,y)$ on tosi kun alkuperäisillä vapailla muuttujilla, x:llä ja y:llä on mainitut arvot. Siksi $\exists x : kaava(x,y)$ on tosi kun alkuperäisillä vapailla muuttujilla ja y:llä on mainitut arvot, joten $\exists y : \exists x : kaava(x,y)$ on tosi kun alkuperäisillä vapailla muuttujilla on mainitut arvot. Lain todistus suuntaan "$\Rightarrow$" on valmis. Vastakkaisen suunnan voi todistaa samalla tavalla.

Laki $\exists x : kaava \Leftrightarrow kaava$ olettaa, että x ei esiinny vapaana $kaava$:ssa. Se riippuu myös toisesta oletuksta. Tätä jälkimmäistä oletusta ei yleensä mainita erikseen, koska sen ajatellaan olevan aina automaattisesti mukana predikaattilogiikkassa. Aloitamme siihen tutustumisen tarkastelemalla seuraavia neljää virkettä. Niistä kaksi ensimmäistä on arkijärjen mukaan totta:

- Jokainen ihminen on ihminen.
- On olemassa ihminen, joka on ihminen.
- Jokainen lohikäärme on lohikäärme.
- On olemassa lohikäärme, joka on lohikäärme.

Koska lohikäärmeitä ei ole olemassa, saatetaan kolmannen virkkeen totuusarvosta kinastella. Logiikassa se on tosi sillä perusteella, että sille ei ole vastaesimerkkejä. Vastaesimerkki sille olisi lohikäärme, joka ei ole lohikäärme. Sellaisia lohikäärmeitä ei ole olemassa, koska minkäänlaisia lohikäärmeitä ei ole olemassa.

Neljäs virke ei ole totta. Sen ongelma on siinä, että se sivutuotteenaan väittää, että lohikäärmeitä on olemassa. Mutta lohikäärmeitä ei ole olemassa.

Esimerkiksi kaava $\exists x : x = x$ muistuttaa neljättä virkettä jonkin verran. Minkä tahansa puheenaiheen mille tahansa alkiolle x pätee $x = x$. Niinpä jos yksikin alkio on olemassa, niin $\exists x : x = x$ on tosi. Mutta jos puheenaiheessa ei ole yhtään alkiota (kuten jos puheenaiheena on lohikäärmeet), niin $\exists x : x = x$ ei ole tosi, koska se väittää, että jokin alkio on olemassa. Siis $\exists x : x = x$ käyttäytyy eri tavoin riippuen siitä, onko puheenaiheessa yhtään alkiota. Se on esimerkki siitä, että tyhjät puheenaiheet noudattavat jossain määrin eri lakeja kuin muut puheenaiheet.

Logiikan sovelluksissa voidaan tavallisesti luottaa, että puheenaihe ei ole tyhjä. Ihmisiä on olemassa, joten jos puheenaiheena on ihmiset, niin puheenaiheessa on jotakin. Kun ruokakaupassa lasketaan paljonko ostokset yhteensä maksavat, niin harvoin pysähdytään ajattelemaan, että kenties tehdäänkin turhaa työtä, koska kenties lukuja ei olekaan olemassa.

Tyhjä puheenaihe on siis erikoistapaus, joka mutkistaa asioita mutta jota harvoin tarvitaan. Siksi predikaattilogiikassa on tapana asettaa vaatimus, että puheenaiheessa on jotakin. Se on itse asiassa ainoa puheenaihetta koskeva vaatimus, joka predikaattilogiikassa asetetaan automaattisesti. On hyvä tiedostaa, että sen vaatiminen on todellakin vain tapa eikä välttämättömyys. Sen voi jättää vaatimatta, mutta siinä tapauksessa jotkin lait pitää muotoilla monimutkaisemmin.

Tämä tarkoittaa vain, että puheenaihe kokonaisuudessaan ei ole tyhjä. Logiikan sovelluksissa toki hyväksytään, että ei ole olemassa 200-vuotiaita ihmisiä eikä ole olemassa lukua, joka kerrottuna nollalla tuottaa ykkösen. Siksi voi olla, että vaikka puheenaihe ei kaiken kaikkiaan ole tyhjä, sen kiinnostava osa on. Tällainen tilanne tulee esiintymään sivulla 51.

Olemme nyt valmiit todistamaan lain $\exists x : \mathit{kaava} \Leftrightarrow \mathit{kaava}$, missä x ei esiinny vapaana *kaava*:ssa. Todistus on lyhyt ja tylsä. Koska puheenaihe ei ole tyhjä, voidaan x:lle valita jokin arvo. Oikean puolen vapaat muuttujat ovat samat kuin vasemman puolen. Tarkastelemme mitä tahansa niiden arvojen yhdistelmää. Jos oikea puoli on sillä tosi, niin myös vasen on, sillä x:lle voidaan valita jokin arvo, ja valittu arvo ei vaikuta siihen, onko *kaava* tosi. Jos vasen puoli on sillä tosi, niin *kaava* on sillä tosi riippumatta x:n arvosta, joten oikea puoli on tosi. Siksi $\exists x : \mathit{kaava} \Leftrightarrow \mathit{kaava}$.

Tälläkin kertaa saattaa olla hyödyllistä miettiä, miten juuri todistettu laki eroaa samankaltaisesta, jossa lakiehdokas ei pidäkään paikkaansa. Täällä voit kokeilla, minkälaisen virheilmoituksen MathCheck antaa epäpätevästä yhtäpitävyydestä $\exists x : x > y \Leftrightarrow x > y$, ja tehdä muutaman jatkotehtävän.

Edellä $\exists x : \mathit{kaava}$ on hiukan kummallinen sikäli, että siinä luodaan kvanttorilla muuttuja mutta ei kuitenkaan käytetä sitä. Se on vähän samankaltainen kuin virke ”on olemassa sellainen ihminen, että Suomi itsenäistyi vuonna 1917.”

On kuitenkin osoittautunut, että sellaiset kaavat eivät ole logiikalle ongelma, ja jos ne yritettäisiin estää, niin esto olisi helppo kiertää. Esimerkiksi ”on olemas-

sa sellainen ihminen, että Suomi itsenäistyi vuonna 1917 ja hän syntyi joko sinä vuonna tai muulloin" käyttää kvanttorilla "on olemassa" luotua muuttujaa "hän", mutta sanoo silti saman kuin "on olemassa sellainen ihminen, että Suomi itsenäistyi vuonna 1917." Ne molemmat sanovat saman kuin "Suomi itsenäistyi vuonna 1917 ja on olemassa ihminen" (joka on tosi).

Todistus $\exists x : \big(kaava1(x) \wedge kaava2(x)\big) \Leftarrow \big(\exists x : kaava1(x)\big) \wedge \big(\exists x : kaava2(x)\big)$ ei mennyt edellä läpi, koska vasemmalla puolella tarvitaan x:n arvo joka toteuttaa sekä $kaava1(x)$:n että $kaava2(x)$:n, mutta oikea puoli lupaa toteuttavan x:n arvon vain kummallekin osakaavalle erikseen, eikä että sama arvo toteuttaa molemmat. Tilanne muuttuu, jos lisätään oletus, että x ei esiinny vapaana $kaava1$:ssä. Oletuksen vallitessa olisi harhaanjohtavaa näyttää $kaava1$:n perässä "(x)", joten jätämme sen pois. Tutustu muuttuneeseen tilanteeseen tällä tehtäväryhmällä ja jatka sitten lukemista.

Oletetaan siis, että x ei esiinny vapaana $kaava1$:ssä. Niinpä jos $kaava1$ toteutuu millään x:n arvolla, niin se toteutuu millä tahansa x:n arvolla. Siksi jos oikea puoli on tosi, niin $kaava1$ toteutuu jokaisella ja $kaava2(x)$ toteutuu jollakin x:n arvolla. Silloin $kaava1 \wedge kaava2(x)$ toteutuu sillä x:n arvolla jolla $kaava2(x)$ toteutuu, joten myös $\exists x : \big(kaava1 \wedge kaava2(x)\big)$ on tosi.

Niinpä jos x ei esiinny vapaana $kaava1$:ssä, niin $\exists x : \big(kaava1 \wedge kaava2(x)\big) \Leftarrow \big(\exists x : kaava1\big) \wedge \big(\exists x : kaava2(x)\big)$. Vastakkainen suunta on todistettu (vähemmillä oletuksilla) jo aiemmin, joten $\exists x : \big(kaava1 \wedge kaava2(x)\big) \Leftrightarrow \big(\exists x : kaava1\big) \wedge \big(\exists x : kaava2(x)\big)$. Edellä todistimme $\exists x : kaava1 \Leftrightarrow kaava1$. (Siellä $kaava1$:n tilalla oli $kaava$, mutta kumpikin edustaa mitä tahansa kaavaa jossa x ei esiinny vapaana.) Siksi

> Jos x ei esiinny vapaana $kaava1$:ssä, niin
> $\exists x : \big(kaava1 \wedge kaava2(x)\big) \;\Leftrightarrow\; kaava1 \wedge \exists x : kaava2(x)$.

Tämän lain voi perustella myös tarkastelemalla erikseen mitä tahansa mahdollista tilannetta, jossa $kaava1$ tuottaa F, ja mitä tahansa mahdollista tilannetta, jossa se tuottaa T. Ensin mainitussa vasen puoli sievenee ensin muotoon $\exists x : \mathsf{F}$ ja siitä edelleen muotoon F. Oikea puoli sievenee heti muotoon F. Jälkimmäisessä tilanteessa kumpikin puoli sievenee heti muotoon $\exists x : kaava2(x)$.

Tässä tehtävässä käskettiin korjata $\exists x : n = 7x \wedge m = 7x$ sanomaan, että sekä n että m on jaollinen seitsemällä. Vastauksiksi kelpasivat sekä $(\exists x : n = 7x) \wedge \exists x : m = 7x$ että $\exists x : n = 7x \wedge \exists x : m = 7x$. Palautteessaan jälkimmäiseen MathCheck näytti osan x:istä eri värillä, mutta siitä pääsi eroon vaihtamalla niiden nimeksi vaikkapa y, eli $\exists x : n = 7x \wedge \exists y : m = 7y$ kelpasi eikä aiheuttanut eri väriä. Mutta $(\exists x : n = 7x) \wedge \exists x : m = 7x$ ja $\exists x : n = 7x \wedge \exists x : m = 7x$ ovat aivan eri kaavoja.

Osuus $m = 7x$ on ensimmäisessä yhden mutta jälkimmäisessä kahden $\exists$:n vaikutuspiirissä. Miten on mahdollista, että molemmat kelpasivat vastauksiksi?

Selitys on edellä todistamassamme laissa; siinä, että x ei esiinny vapaana kaavassa $(\exists x : m = 7x)$; sekä siinä, että $kaava1 \wedge kaava2 \Leftrightarrow kaava2 \wedge kaava1$ on laki. Niiden ansiosta

$$
\begin{array}{lll}
 & \exists x : n = 7x \wedge \exists x : m = 7x & \\
\Leftrightarrow & \exists x : (n = 7x \wedge (\exists x : m = 7x)) & \text{lisätty tarpeettomia sulkeita} \\
\Leftrightarrow & \exists x : ((\exists x : m = 7x) \wedge n = 7x) & \wedge\text{:n osakaavat vaihdettu keskenään} \\
\Leftrightarrow & (\exists x : m = 7x) \wedge (\exists x : n = 7x) & \text{edellä todistettu laki} \\
\Leftrightarrow & (\exists x : n = 7x) \wedge (\exists x : m = 7x) & \wedge\text{:n osakaavat vaihdettu keskenään} \\
\Leftrightarrow & (\exists x : n = 7x) \wedge \exists x : m = 7x & \text{poistettu tarpeettomat sulkeet}
\end{array}
$$

Olemassaolokvanttorista on hyvä tietää vielä yksi asia: jos kvantifioidulla muuttujalla voi olla vain äärellinen määrä eri arvoja, niin ”$\exists$” voidaan korvata ”$\vee$”:lla. Tämä säilyy voimassa, jos muuttujalla voi olla äärettömästi eri arvoja, mutta niistä vain äärellisellä määrällä voi kvanttorin alainen kaava tuottaa muun totuusarvon kuin F. Esimerkiksi kokonaisluvuista puhuttaessa $\exists i : 2 \leq i \leq 5 \wedge kaava(i)$ tarkoittaa samaa kuin $kaava(2) \vee kaava(3) \vee kaava(4) \vee kaava(5)$. Siitä on yksi tehtävä täällä.

3.2 Yhtäjonkin käyttö kaavan sisällä

Kun edellä päättelimme $\exists x : (n = 7x \wedge (\exists x : m = 7x)) \Leftrightarrow \exists x : ((\exists x : m = 7x) \wedge n = 7x)$, niin sovelsimme lakia $kaava1 \wedge kaava2 \Leftrightarrow kaava2 \wedge kaava1$ isomman kaavan $\exists x : (kaava1 \wedge kaava2)$ sisällä siten, että $kaava1$ oli $n = 7x$ ja $kaava2$ oli $(\exists x : m = 7x)$. Osakaavan korvaaminen yhtäpitävällä on erittäin tehokas päättelykeino. Mutta ikävä kyllä, se on myös sikäli vaikea, että ei ole aina helppo nähdä, saako sitä käyttää. Tässä luvussa käsittelemme tapauksia, joissa sitä saa käyttää. Edellä ollut käyttö saa oikeutuksen. Tarkastelemme myös lausekkeen korvaamista yhtäsuurella — tai ”laajennetusti yhtäsuurella” — kaavan sisällä. Näitä ennen kuitenkin tutustumme erääseen vaikeuteen logiikan soveltamisessa käytäntöön.

Logiikan soveltamisessa matematiikkaan, ohjelmointiin ja arkielämään tulee vastaan ilmiö, johon logiikan valtavirta ei ole varautunut: määrittelemättömät ilmaukset. Seuraava esimerkki havainnollistaa niiden muodostamaa ongelmaa. Koska nollalla jakaminen ei ole määriteltyä, ei kumpaakaan kaavoista $\frac{1}{0} < 0$ ja $\frac{1}{0} \geq 0$ voi hyväksyä todeksi. Jollei ole muita totuusarvoja kuin ”tosi” ja ”epätosi”, on molempien oltava epätosi. Reaaliluvuilla pätee $a \geq b \Leftrightarrow \neg(a < b)$. Jos lauseketta $\frac{1}{0}$ voisi käsitellä kuten mitä tahansa reaalilukua, niin voitaisiin päätellä $\frac{1}{0} \geq 0 \Leftrightarrow \neg(\frac{1}{0} < 0)$. Se on ristiriidassa sen kanssa, että sekä $\frac{1}{0} < 0$ että $\frac{1}{0} \geq 0$ on epätosi.

$\neg$	
F	T
U	U
T	F

$\wedge$	F	U	T
F	F	F	F
U	F	U	U
T	F	U	T

$\vee$	F	U	T
F	F	U	T
U	U	U	T
T	T	T	T

Kuva 3: Totuusarvojen käyttäytyminen

Logiikan valtavirrassa tämä ongelma on sysätty syrjään. Siellä oletetaan, että jokainen laskutoimitus tai muu sen kaltainen tuottaa aina puheenaiheen piiriin kuuluvan arvon. Mutta nollalla jakaminen ei tuota. Siksi jakolaskua ei voi esittää sellaisenaan tavanomaisessa logiikassa! Sama koskee esimerkiksi luonnollisen kielen ilmausta ”henkilön puoliso”, sillä on ihmisiä, joilla ei ole puolisoa. Määrittelemättömien ilmausten sallimista on tutkittu paljon logiikassa ja tietojenkäsittelytieteessä, mutta mikään ehdotus ei ole edennyt osaksi valtavirtaa.

Siksi logiikan valtavirrassa jätetään jakolasku tavallisesti pois sallittujen symbolien joukosta. Jakolaskun sijaan sanotaan, että on olemassa sellainen luku, että kun sen kertoo jakajalla niin saadaan jaettava, ja sillä on se ominaisuus mikä osamäärällä haluttiin olevan. Tarvittaessa lisätään tieto, mikä totuusarvon tulee olla kun jakaja on nolla. Siis kaavojen tyyppiä $kaava\left(\frac{a}{b}\right)$ sijaan kirjoitetaan tyyliin $b \neq 0 \wedge \exists c : cb = a \wedge kaava(c)$ tai $b = 0 \vee \exists c : cb = a \wedge kaava(c)$. Tämä on kömpelöä. Lisäksi toisinaan on vaikea päättää, pitäisikö kirjoittaa ”$b \neq 0 \wedge$” vai ”$b = 0 \vee$”.

MathCheckissä tämä ongelma on ratkaistu ottamalla käyttöön kolmas totuusarvo U tarkoittamaan, että kaavan totuusarvo on määrittelemätön (undefined). Sen käyttäytyminen ja muut yksityiskohdat on pyritty valitsemaan siten, että ne mahdollisimman pitkälle vastaisivat sitä, miten määrittelemätöntä tavallisesti käsitellään matematiikassa. Osa käyttäytymisestä on esitetty kuvassa 3. Vertailu tuottaa U jos ja vain jos ainakin yksi verrattavista on määrittelemätön. Esimerkiksi $\frac{a}{b} < 0$, $\neg(\frac{a}{b} < 0)$, $\frac{a}{b} \geq 0$ ja jopa $\frac{a}{b} = \frac{a}{b}$ tuottavat U kun $b = 0$, mutta $b \neq 0 \wedge \frac{a}{b} < 0$ on epätosi ja $b = 0 \vee \frac{a}{b} < 0$ on tosi kun $b = 0$. MathCheckissä $\neg(lauseke1 < lauseke2)$ tarkoittaa samaa kuin $lauseke1 \geq lauseke2$ myös silloin, kun jompikumpi tai molemmat lausekkeet ovat määrittelemättömiä.

Kaava $\exists x : kaava(x)$ tuottaa U jos ja vain jos $kaava(x)$ tuottaa U ainakin yhdellä eikä tuota T millään x:n arvolla. Kaava $\forall x : kaava(x)$ tuottaa U jos ja vain jos $kaava(x)$ tuottaa U ainakin yhdellä eikä tuota F millään x:n arvolla.

Koska nämä asiat eivät kuulu logiikan valtavirtaan, niistä puhutaan tässä tekstissä mahdollisimman vähän. Niistä on kuitenkin pakko puhua jonkin verran siksi, että ne ilmenevät MathCheckiä käytettäessä. Myös on hyödyllistä tutustua asioihin, jotka auttavat käsittelemään määrittelemätöntä oikein ohjelmoinnissa ja matematiikassa. Sitä varten on muutamia tehtäviä. Niistä ensimmäiset ovat täällä.

Tehtävissä tarkasteltiin seuraavaa yhtälön ratkaisua:

$$\begin{aligned}\frac{x+1}{x-1} = 2 \quad &\Leftrightarrow \quad x-1 \neq 0 \ \wedge \ x+1 = 2(x-1) \\ &\Leftrightarrow \quad x \neq 1 \ \wedge \ x+1 = 2x-2 \\ &\Leftrightarrow \quad x \neq 1 \ \wedge \ 3 = x \\ &\Leftrightarrow \quad 3 = x \quad \Leftrightarrow \quad x = 3\end{aligned}$$

Ensimmäisessä vaiheessa kerrottiin yhtälön molemmat puolet $(x-1)$:llä ja lisättiin "$x-1 \neq 0 \wedge$". Tämä lisäys tehtiin varmistamaan, että lopputulos ei väitä, että x voi olla 1. Se ei saa olla 1, koska alkuperäinen yhtälö ei ole määritelty kun $x = 1$, koska silloin siinä jaetaan nollalla. Kun $x = 1$, on ensimmäisen "$\Leftrightarrow$":n vasen puoli määrittelemätön ja oikea puoli epätosi. Niinpä kun ensimmäisen "$\Leftrightarrow$":n jompikumpi puoli on tosi, on $x \neq 1$. Silloin jako- ja kertolaskun käänteisyydestä seuraa, että toinenkin puoli on tosi. Sivun 6 mukaan "$\Leftrightarrow$" vaatii "molemmat tosi tai ei kumpikaan tosi". Se ei vaadi, että puolet saavat saman totuusarvon silloin kun kumpikaan puoli ei ole tosi. Siksi ensimmäinen "$\Leftrightarrow$" on pätevä.

Toisessa vaiheessa sievennettiin $x-1 \neq 0 \Leftrightarrow x \neq 1$ ja $x+1 = 2(x-1) \Leftrightarrow x+1 = 2x-2$. Sitten $x+1 = 2x-2$ muutettiin muotoon $3 = x$ lisäämällä molemmille puolille $2-x$. Nämä tehtiin isomman kaavan sisällä, eli esimerkiksi siitä, että $x+1 = 2x-2 \Leftrightarrow 3 = x$ pääteltiin $x \neq 1 \wedge x+1 = 2x-2 \Leftrightarrow x \neq 1 \wedge 3 = x$. Lupa sievennöksien tekemiseen kaavan sisällä tulee myöhemmin tässä luvussa esitettävistä päättelysäännöistä.

Tässä vaiheessa näkyy, että x voi olla vain 3, joten "$x \neq 1 \wedge$" jätettiin tarpeettomana pois. Lopuksi $3 = x$ käännettiin toisinpäin, koska yhtälöiden ratkaisut on tapana ilmoittaa niinpäin.

Tehtävässä kokeiltiin, miten palaute muuttuu, kun "$x-1 \neq 0 \wedge$" jätetään pois. Se jäi tietenkin pois palautteestakin, mutta muuten palaute ei muuttunut. Kun $x \neq 1$, on $x-1 \neq 0$ tosi, joten $x-1 \neq 0 \wedge x+1 = 2(x-1)$ tuottaa saman totuusarvon kuin $x+1 = 2(x-1)$. Ne tuottavat saman totuusarvon myös kun $x = 1$, koska silloin molemmat ovat epätosi, koska $x+1 = 2$ ja $2(x-1) = 0$, joten $x+1 = 2(x-1)$ on epätosi. Poisto ei siis vaikuttanut toisen kaavan tuottamiin totuusarvoihin, joten se ei myöskään vaikuttanut koko päättelyketjun pätevyyteen.

Vaatimusta, että jakaja ei saa olla nolla, ei siis ole aina tarpeen lisätä. Jos sen jättää lisäämättä kun se on tarpeen niin tekee päättelyvirheen. Jos sen lisää kun se ei ole tarpeen niin tekee ehkä turhaa työtä, mutta ei tee päättelyvirhettä. Siksi se kannattaa laittaa mukaan aina kun ei ole varma, että sen voi jättää pois.

Lopuksi tehtävässä havainnollistettiin sitä, että kun MathCheck tarkastaa yhtälön, epäyhtälön tai niistä muodostetun ryhmän ratkaisun, se vaatii niin hyvin kuin osaa, että lopputulos esitetään loppuun saakka sievennettynä ilman mitään turhaa.

Seuraavilla tehtävillä voi harjoitella määrittelemätöntä yhtälöiden ratkaisemisessa. Sen sijaan että yrittää kirjoittaa heti lopullisen vastauksen, kannattaa edetä vai-

he vaiheelta. Vaiheet erotetaan toisistaan "$\Leftrightarrow$":llä. Vaiheet kannattaa valita itselle sopiviksi: jos osaa asian hyvin, niin voi edetä pitkin askelin, mutta jos on epävarma pikkuasioissakin, kannattaa edetä yksityiskohta kerrallaan. Koska vastauksen osia voi maalata ja kopioida, ei kirjoittamisvaiva ole kohtuuton. Kukin vaihe kannattaa miettiä hyvin, kirjoittaa se vastauslaatikkoon ja tarkastuttaa samantien MathCheckillä. Jos vaihe ei ollut oikein, niin kannattaa palautteen perusteella yrittää ymmärtää miksi se oli väärin, eikä vain arvata jokin korjaus. Tällainen on tehokas oppimismenetelmä.

Vaatimus, että jakaja ei ole nolla, oli välttämätön toisessa tehtävässä. Jos yritit jättää sen pois, niin MathCheck (toivottavasti!) huomautti heti.

Kolmannessa tehtävässä harjoiteltiin sekä päättelyn jakamista tapauksiin että sitä, että epäyhtälön suunta muuttuu, kun sen molemmat puolet kerrotaan negatiivisella luvulla. Kun jakaja on negatiivinen, on oletuksena $x < 4$ ja ratkaisuksi tulee $x \leq 7$. Niistä yhdessä tulee $x < 4 \wedge x \leq 7$, joka sievenee muotoon $x < 4$. Kun jakaja on positiivinen, on oletuksena $x > 4$ ja ratkaisuksi tulee $x \geq 7$. Niistä yhdessä tulee $x > 4 \wedge x \geq 7$, joka sievenee muotoon $x \geq 7$. Ratkaisujen yhdistäminen tuottaa $x < 4 \vee x \geq 7$.

Oletuksen $x < 4$ alaisuudessa todellakin saa sieventää $x \leq 7 \Leftrightarrow x < 4$. Edellä määriteltiin "$\Leftrightarrow$" niin, että jokaisessa mahdollisessa tilanteessa, jossa toinen puoli on tosi, myös vastakkainen puoli on tosi. "Tilanne" tarkoittaa vapaiden muuttujien arvojen yhdistelmää. "Mahdollinen" määräytyy asiayhteydessä voimassa olevista oletuksista. Kun on tehty oletus $x < 4$, niin esimerkiksi tilanne $x = 2$ on mahdollinen, mutta tilanne $x = 6$ ei ole mahdollinen. Siis oletuksen $x < 4$ alaisuudessa on väliä vain sillä, minkä totuusarvon kaava tuottaa, kun $x < 4$. Kun $x < 4$, tuottaa $x \leq 7$ aina saman totuusarvon kuin $x < 4$, joten $x \leq 7 \Leftrightarrow x < 4$.

Tehtäväryhmän viimeisessä tehtävässä kiinnitettiin huomiota siihen, että MathCheck saattaa hyväksyä väärin ajatellun vastauksen, jos se sattuu olemaan matemaattisesti yhtäpitävä oikean vastauksen kanssa. Oletuksen $x < 4$ alaisuudessa oltaessa jokainen kaava muotoa $x < \mathit{luku}$ tai $x \leq \mathit{luku}$, missä $\mathit{luku} > 4$, tarkoittaa samaa kuin $x < 4$. MathCheck hyväksyy ne kaikki. Tästä syystä (ja muista syistä) MathCheck ei paljasta kaikkia päättelyvirheitä. Mutta kun tehtäviä ratkaisee riittävän monta, niin jollei väärä ajatusmalli paljastu ensimmäisen tehtävän kohdalla, se saattaa hyvinkin paljastua jonkin myöhemmän tehtävän kohdalla.

Kuten edellä on todettu, MathCheckissä $\mathit{kaava}1 \Leftrightarrow \mathit{kaava}2$ tarkoittaa, että jokaisessa mahdollisessa tilanteessa, jossa $\mathit{kaava}1$ on tosi, myös $\mathit{kaava}2$ on tosi, ja päinvastoin. Nyt kerrotaan, että MathCheckissä $\mathit{kaava}1 \equiv \mathit{kaava}2$ tarkoittaa, että jokaisessa mahdollisessa tilanteessa $\mathit{kaava}1$ tuottaa saman totuusarvon kuin $\mathit{kaava}2$. Sanomme, että $\mathit{kaava}1$ ja $\mathit{kaava}2$ ovat *yhtätodet* ja "$\equiv$" on *yhtätotuus*. Näiden kahden ero on siinä, että $\mathit{kaava}1 \Leftrightarrow \mathit{kaava}2$ sallii mutta $\mathit{kaava}1 \equiv \mathit{kaava}2$ ei salli, että toinen kaavoista on epätosi ja toinen määrittelemätön. Tästä asiasta

on yksi tehtävä.

Kumpikin näistä merkinnöistä tarvitaan erikseen. Yhtälöiden ratkaisemisessa ”$\Leftrightarrow$” eli yhtäpitävyys on parempi, koska esimerkiksi $\frac{x+1}{x-1} = 2 \Leftrightarrow x = 3$ on pätevä, mutta $\frac{x+1}{x-1} = 2 \equiv x = 3$ ei ole pätevä, sillä kun $x = 1$, on sen vasen puoli määrittelemätön mutta oikea puoli epätosi. Toisaalta kun pääteltäessä halutaan korvata kaavan osakaava toisella, niin ”$\equiv$” eli yhtätotuus on parempi. Siirrymme nyt käsittelemään osakaavan korvaamista.

Sivulla 18 kerrottiin, että tiettyjen rajoitusten vallitessa $kaava1 \Leftrightarrow kaava2$ takaa $isompi\_kaava(kaava1) \Leftrightarrow isompi\_kaava(kaava2)$. Nyt on aika kertoa ne rajoitukset. Niitä on kaksi:

R1 Mikään $kaava1$:n tai $kaava2$:n vapaa muuttuja, joka esiintyy vapaana myös voimassa olevissa oletuksissa, ei saa joutua sidotuksi $isompi\_kaava(kaava1)$:ssä eikä $isompi\_kaava(kaava2)$:ssa.

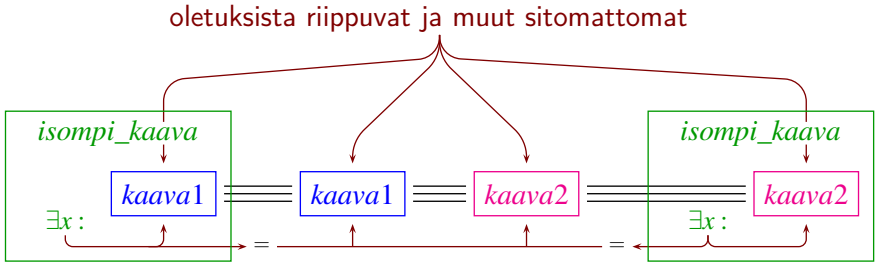
R2 Jokaisessa mahdollisessa tilanteessa, jossa $kaava1$ tuottaa määrittelemättömän totuusarvon, myös $kaava2$ tuottaa määrittelemättömän totuusarvon, ja päinvastoin.

R2 ja $kaava1 \Leftrightarrow kaava2$ tarkoittavat yhdessä samaa kuin $kaava1 \equiv kaava2$, eli että jokaisessa mahdollisessa tilanteessa $kaava1$ tuottaa saman totuusarvon kuin $kaava2$. Tätä hyödyntämällä päättelysääntö voidaan esittää tiiviimmin:

> **Yhtätotuuden käyttö kaavan sisällä:** Jos $kaava1 \equiv kaava2$ ja jos mikään $kaava1$:n tai $kaava2$:n vapaa muuttuja, joka esiintyy vapaana myös voimassa olevissa oletuksissa, ei joudu sidotuksi $isompi\_kaava(kaava1)$:ssä eikä $isompi\_kaava(kaava2)$:ssa, niin $isompi\_kaava(kaava1) \equiv isompi\_kaava(kaava2)$.

Tämä päättelysääntö lupaa hieman enemmän kuin sivulla 18 luvattiin: siellä luvattiin vain $isompi\_kaava(kaava1) \Leftrightarrow isompi\_kaava(kaava2)$, mutta nyt luvataan lisäksi, että $isompi\_kaava(kaava1)$ tuottaa määrittelemättömän totuusarvon silloin ja vain silloin kun $isompi\_kaava(kaava2)$ tuottaa määrittelemättömän totuusarvon. Mitä enemmän luvataan, sen parempi, koska luvatuista asioista saattaa olla hyötyä päättelyiden myöhemmissä vaiheissa, mutta niitä ei ole pakko käyttää jollei niitä tarvitakaan.

R1:n tärkeyden ymmärtämiseksi on tarpeen muistaa, että päättely tapahtuu usein joidenkin oletusten alaisuudessa. Tässä yhteydessä oletus voi tarkoittaa myös aiemmin saatua tietoa, joka ei ole kirjoitettu uuden päättelyn lähtökohdaksi. Esimerkiksi jos yhtälöparia ratkaistaessa on saatu selville, että $x = 4$, niin on oikein ja hyödyllistä päätellä $y = 2x + 3 \Leftrightarrow y = 2 \cdot 4 + 3$. Tämä yhtäpitävyys ei tietenkään ole pätevä kaikissa tilanteissa, mutta on pätevä kaikissa niissä tilanteissa,



Kuva 4: *isompi_kaava*(*kaava*1) ≡ *isompi_kaava*(*kaava*2)

jotka ovat mahdollisia, kun $x = 4$. Mutta jos *isompi_kaava*(*kaava*) on $\exists x : kaava$, niin ilman R1:tä saataisiin $\exists x : y = 2x + 3 \Leftrightarrow \exists x : y = 2 \cdot 4 + 3$. Se ei ole pätevä, koska sen vasen puoli sanoo että y on mikä tahansa luku, mutta oikea puoli sanoo että $y = 11$.

R2 estää muun muassa päättelemästä pätevästä yhtäpitävyydestä $x < \frac{1}{0} \Leftrightarrow \mathsf{F}$ epäpätevän yhtäpitävyyden $\neg(x < \frac{1}{0}) \Leftrightarrow \neg\mathsf{F}$. Se on epäpätevä, koska $\neg(x < \frac{1}{0})$ ei ole tosi vaan määrittelemätön, ja MathCheckissä on tietysti pidetty kiinni siitä, että $\neg\mathsf{F}$ tuottaa T eikä U.

Helpottaaksemme R1:stä puhumista otamme tilapäisesti käyttöön käsitteen ”oletuksista riippuva”. Vapaana *kaava*1:ssä tai *kaava*2:ssa esiintyvä muuttuja on oletuksista riippuva, jos ja vain jos se esiintyy vapaana myös voimassa olevissa oletuksissa. R1 varmistaa, että oletuksista riippuva muuttuja ei voi vaihtua toiseen samannimiseen. Siksi *kaava*1 ja *kaava*2 saavat oletuksista riippuviin muuttujiin samat arvot *isompi_kaava*:n sisällä kuin irrallaan. Sama pätee myös niihin *kaava*1:n ja *kaava*2:n vapaisiin muuttujiin, jotka eivät riipu oletuksista ja joita ei sidota *isompi_kaava*:ssa. Kuvassa 4 tätä on havainnollistettu sanoista ”oletuksista riippuvat ja muut sitomattomat” alkavilla nuolilla.

Kolmas ja samalla viimeinen ryhmä *kaava*1:n ja *kaava*2:n vapaita muuttujia on ne, jotka joutuvat sidotuiksi *isompi_kaava*:ssa. Jos kullekin niistä annetaan *isompi_kaava*:n ulkopuolella sama arvo kuin *isompi_kaava*:n sisällä, niin *kaava*1 tuottaa *isompi_kaava*:n sisällä saman totuusarvon kuin irrallaan, ja samoin *kaava*2:lle. Kuvan 4 alaosan nuolet ja laitimmaiset kolmoisviivat havainnollistavat tätä.

Jos tilanne on mahdollinen, niin se säilyy mahdollisena kun kolmannen ryhmän muuttujan arvo vaihtuu, koska R1:n vuoksi muuttuja ei riipu oletuksista. Lähtökohdan *kaava*1 ≡ *kaava*2 ansiosta *kaava*1 tuottaa saman totuusarvon kuin *kaava*2, onpa kolmannen ryhmän muuttujilla mitkä tahansa arvot. (Tämä yhteinen totuusarvo voi vaihdella kolmannen ryhmän muuttujien arvojen vaihdellessa.) Kuvan 4 keskimmäinen kolmoisviiva havainnollistaa tätä. Siitä seuraa (käymme tämän kohta tarkemmin läpi), että *isompi_kaava*(*kaava*1) tuottaa saman totuusar-

von kuin $isompi\_kaava(kaava2)$.

Osoittaaksemme päättelysäännön ”Yhtätotuuden käyttö kaavan sisällä” päteväksi, otamme lähtökohdiksi (1) $kaava1 \equiv kaava2$ ja (2) mikään $kaava1$:n tai $kaava2$:n vapaa muuttuja, joka esiintyy vapaana myös voimassa olevissa oletuksissa, ei joudu sidotuksi $isompi\_kaava(kaava1)$:ssä eikä $isompi\_kaava(kaava2)$:ssa. Osoitamme $isompi\_kaava(kaava1) \equiv isompi\_kaava(kaava2)$.

Tarkastelemme ensin yhden $kaava1$:n esiintymän korvaamista $kaava2$:lla. Aloitamme osoittamalla, että on olemassa sellainen kaava $ik(X)$, että mikään oletusten vapaa muuttuja ei esiinny siinä sidottuna, $isompi\_kaava(kaava1) \equiv ik(kaava1)$, ja $isompi\_kaava(kaava2) \equiv ik(kaava2)$.

Tarkoittakoon $isompi\_kaava(X)$ kaavaa, joka on muuten sama kuin $isompi\_kaava(kaava1)$, mutta jossa korvattavan esiintymän tilalla on X. Siitä tehdään $ik(X)$ vaihtamalla jokainen sidotun muuttujan nimi, joka on sama kuin oletusten jonkin vapaan muuttujan nimi, nimeen, joka ei ole minkään muun muuttujan nimi. Kuten sivulla 28 todettiin, tällaiset vaihdot eivät muuta kaavan merkitystä. Muutokset eivät vaikuta X:n kohdalla niihin nimiin, jotka esiintyvät vapaina $kaava1$:ssä tai $kaava2$:ssa, koska (2):n mukaan sellainen nimi ei esiinny vapaana oletuksissa tai ei ole sidottu X:n kohdalla. Siksi $isompi\_kaava(kaava1) \equiv ik(kaava1)$ ja $isompi\_kaava(kaava2) \equiv ik(kaava2)$.

Seuraavaksi täytyy osoittaa, että $ik(kaava1) \equiv ik(kaava2)$. Olkoon K_1 mikä tahansa $ik(kaava1)$:n osakaava joka sisältää $kaava1$:n korvattavan esiintymän, ja olkoon K_2 vastaava $ik(kaava2)$:n osakaava. Pienimmillään K_1 on sama kuin $kaava1$ ja suurimmillaan se on $ik(kaava1)$, ja samoin K_2:lle. Seuraavaksi osoitamme, että jos $K_1 \equiv K_2$ pätee joillekin K_1 ja K_2, niin se pätee myös seuraavaksi suuremmille K_1 ja K_2. Teemme sen osoittamalla, että jos $K_1 \equiv K_2$, niin

$$\begin{aligned}
\neg K_1 &\equiv \neg K_2 \\
K_1 \wedge muu &\equiv K_2 \wedge muu \\
muu \wedge K_1 &\equiv muu \wedge K_2 \\
K_1 \vee muu &\equiv K_2 \vee muu \\
muu \vee K_1 &\equiv muu \vee K_2 \\
\forall x : K_1 &\equiv \forall x : K_2 \\
\exists x : K_1 &\equiv \exists x : K_2
\end{aligned}$$

Tarkastelemme mitä tahansa mahdollista tilannetta. Ensimmäinen rivi seuraa suoraan siitä, että $\neg kaava$:n tuottama totuusarvo riippuu vain $kaava$:n tuottamasta totuusarvosta, eikä riipu $kaava$:n mistään muista ominaisuuksista. Lähtökohdan $K_1 \equiv K_2$ ansiosta K_1 tuottaa tarkasteltavassa tilanteessa saman totuusarvon kuin K_2, joten $\neg K_1$ tuottaa siinä tilanteessa saman totuusarvon kuin $\neg K_2$. Myöskään ”$\wedge$”:n ja ”$\vee$” tuottamat totuusarvot eivät riipu muusta kuin osakaavojen tuottamista totuusarvoista, joten myös neljä seuraavaa riviä pätee.

Kahden alimman rivin x ei esiinny vapaana oletuksissa, koska mikään oletusten vapaa muuttuja ei esiinny sidottuna $ik(X)$:ssä. Siksi mahdollinen tilanne säilyy mahdollisena, jos x:n arvo vaihdetaan. Lähtökohdan $K_1 \equiv K_2$ ansiosta K_1 tuottaa millä tahansa x:n arvolla ja tarkasteltavassa tilanteessa vallitsevalla muiden muuttujien arvojen yhdistelmällä saman totuusarvon kuin K_2. Siksi kaksi alinta riviä pätee.

Nyt voimme aloittaa siten, että K_1 ja K_2 ovat $kaava1$ ja $kaava2$, ja edetä askel kerrallaan isompiin osakaavoihin kunnes K_1 ja K_2 ovat $ik(kaava1)$ ja $ik(kaava2)$. (1):n ansiosta alussa pätee $K_1 \equiv K_2$. Jokaisessa seuraavassa vaiheessa $K_1 \equiv K_2$ pätee siksi, että se päti edellisessä vaiheessa ja seuraava vaihe rakentuu edellisestä jollakin edellä käsitellyistä tavoista. Lopulta saadaan, että $ik(kaava1) \equiv ik(kaava2)$ pätee.

Jäljellä on vielä tapaukset, joissa nolla tai ainakin kaksi $kaava1$:n esiintymää korvataan $kaava2$:lla. Jos kaksi esiintymää korvataan, niin voimme käsitellä kummankin korvaamisen erikseen. Edellä todistetun perusteella

$$\begin{aligned} isompi\_kaava(kaava1, kaava1) &\equiv isompi\_kaava(kaava1, kaava2) \\ &\equiv isompi\_kaava(kaava2, kaava2) \end{aligned}$$

Sama periaate toimii myös useammalle kuin kahdelle esiintymälle. Jos $kaava1$ ei esiinny kertaakaan $isompi\_kaava(kaava1)$:ssä, niin $isompi\_kaava(kaava1)$ tuottaa saman totuusarvon kuin $isompi\_kaava(kaava2)$, koska ne ovat täysin sama kaava. Päättelysäännön ”Yhtätotuuden käyttö kaavan sisällä” todistus on valmis.

Luvussa 3.1 siitä, että $n = 7x \wedge (\exists x : m = 7x) \Leftrightarrow (\exists x : m = 7x) \wedge n = 7x$, pääteltiin $\exists x : (n = 7x \wedge (\exists x : m = 7x)) \Leftrightarrow \exists x : ((\exists x : m = 7x) \wedge n = 7x)$. Tässä esimerkissä ei esiinny nollalla jakamista eikä muitakaan määrittelemättömiä toimenpiteitä, joten ”$\equiv$” ja ”$\Leftrightarrow$” eivät eroa toisistaan, joten niistä saa käyttää kumpaa tahansa. Sikäli ”Yhtätotuuden käyttö kaavan sisällä” olisi siis käytettävissä. Mutta koska osakaavan $n = 7x$ vapaana sisältämä x on sidottu isommassa kaavassa, ei ”Yhtätotuuden käyttö kaavan sisällä” sellaisenaan oikeuta tätä johtopäätöstä, jos x esiintyy vapaana myös voimassa olevissa oletuksissa.

Tämä este on kuitenkin helppo kiertää. Valitaan y:ksi muuttuja, joka ei esiinny vapaana voimassa olevissa oletuksissa. Yhtäpitävyys $n = 7x \wedge (\exists x : m = 7x) \Leftrightarrow (\exists x : m = 7x) \wedge n = 7x$ saatiin luvussa 3.1 siitä, että $kaava1 \wedge kaava2 \Leftrightarrow kaava2 \wedge kaava1$ on laki. Mutta samasta syystä $n = 7y \wedge (\exists x : m = 7x) \Leftrightarrow (\exists x : m = 7x) \wedge n = 7y$. Näin ollen ”Yhtätotuuden käyttö kaavan sisällä” oikeuttaa alla olevan päättelyketjun toisen askeleen. Ensimmäinen ja kolmas askel seuraavat siitä, että sidotun muuttujan nimen saa valita vapaasti kunhan se ei sekaannu muiden muuttujien nimiin (sivu 28).

$$
\begin{aligned}
& \exists x : (n = 7x \wedge (\exists x : m = 7x)) \\
\Leftrightarrow\ & \exists y : (n = 7y \wedge (\exists x : m = 7x)) \\
\Leftrightarrow\ & \exists y : ((\exists x : m = 7x) \wedge n = 7y) \\
\Leftrightarrow\ & \exists x : ((\exists x : m = 7x) \wedge n = 7x)
\end{aligned}
$$

Vielä helpompaa on käyttää seuraavaa päättelysääntöä:

> **Yhtätotuuslain käyttö kaavan sisällä:** Jos jokaisessa tilanteessa $kaava1 \equiv kaava2$, niin jokaisessa tilanteessa $isompi\_kaava(kaava1) \equiv isompi\_kaava(kaava2)$.

Tämä päättelysääntö voidaan perustella melkein samoin kuin yhtätotuuden käyttö kaavan sisällä perusteltiin. Ainoa ero liittyy kvanttorien käsittelyyn. Oletetaan, että $K_1 \equiv K_2$ on voimassa jokaisessa tilanteessa. Siis K_2 tuottaa millä tahansa muuttujien arvojen yhdistelmällä saman totuusarvon kuin K_1. Siksi, kun x käy läpi kaikki puheenaiheen arvot, tuottaa K_2 kullekin saman totuusarvon kuin K_1, joten $\forall x : K_2$ ja $\exists x : K_2$ tuottavat samat totuusarvot kuin $\forall x : K_1$ ja $\exists x : K_1$. Niinpä jokaisessa tilanteessa $\forall x : K_1 \equiv \forall x : K_2$ ja $\exists x : K_1 \equiv \exists x : K_2$.

Edellä olevat päättelysäännöt oikeuttavat osakaavan korvaamisen yhtätodella osakaavalla. Seuraavat päättelysäännöt oikeuttavat lausekkeen korvaamisen ”laajennetusti yhtäsuurella” lausekkeella kaavan sisällä. Kaksi lauseketta ovat laajennetusti yhtäsuuret, jos ja vain jos ne tuottavat saman arvon tai molemmat ovat määrittelemättömät. Esimerkiksi lausekkeet $1 + \frac{2}{x-1}$ ja $\frac{x+1}{x-1}$ ovat laajennetusti yhtäsuuret kaikilla x:n arvoilla mutta yhtäsuuret vain kun $x \neq 1$. Laajennetun yhtäsuuruuden käyttö tavallisen yhtäsuuruuden sijaan laajentaa päättelysääntöjen käyttömahdollisuuksia. Se esimerkiksi oikeuttaa sievennöksen kaavan sisällä yhtäsuuruudella $1 + \frac{2}{x-1} = \frac{x+1}{x-1}$, vaikka se ei ole tosi kun $x = 1$.

> **Yhtäsuuruuden käyttö kaavan sisällä:** Jos jokaisessa mahdollisessa tilanteessa $lauseke1$ on laajennetusti yhtäsuuri kuin $lauseke2$, ja jos mikään $lauseke1$:n tai $lauseke2$:n muuttuja, joka esiintyy vapaana voimassa olevissa oletuksissa, ei joudu sidotuksi $isompi\_kaava(kaava1)$:ssä eikä $isompi\_kaava(kaava2)$:ssa, niin $isompi\_kaava(lauseke1) \equiv isompi\_kaava(lauseke2)$.

> **Yhtäsuuruuslain käyttö kaavan sisällä:** Jos jokaisessa tilanteessa $lauseke1$ on laajennetusti yhtäsuuri kuin $lauseke2$, niin jokaisessa tilanteessa $isompi\_kaava(lauseke1) \equiv isompi\_kaava(lauseke2)$.

Nämä päättelysäännöt ovat pätevät hyvin samanlaisista syistä kuin edellä esitetyt yhtätotuuden päättelysäännöt. Jos $lauseke1$ on laajennetusti yhtäsuuri kuin $lauseke2$, niin $lauseke1 + lauseke3$ on laajennetusti yhtäsuuri kuin $lauseke2 +$

lauseke3 ja niin edelleen. Samoin *lauseke1* $<$ *lauseke3* tuottaa saman totuusarvon vaihtoehdoista F, T ja U kuin *lauseke2* $<$ *lauseke3* tuottaa ja niin edelleen. Ja esimerkiksi siitä, että *lauseke1* $<$ *lauseke3* tuottaa yhtäsuuruutta koskevassa päättelysäännössä mainituilla ehdoilla saman totuusarvon kuin *lauseke2* $<$ *lauseke3*, seuraa vastaavalla yhtätotuuden päättelysäännöillä, että *isompi_kaava*(*lauseke1*) $\equiv$ *isompi_kaava*(*lauseke2*).

Taas on aika harjoitella!

Edellä olleiden päättelysääntöjen niin-osissa olevien "$\equiv$" tilalle saa aina kirjoittaa "$\Leftrightarrow$", koska *kaava1* $\Leftrightarrow$ *kaava2* pätee automaattisesti aina kun *kaava1* $\equiv$ *kaava2* pätee.

Jos jokainen käytössä olevista laskutoimituksista, vertailuista ja muista sellaisista tuottaa aina määritellyn lopputuloksen, ja jos kaavoissa ei esiinny niiden lisäksi muita symboleita kuin vakioita, muuttujia, sulkeita, "F", "T", "$\neg$", "$\wedge$", "$\vee$", "$\forall$" ja "$\exists$", niin mikään kaava ei koskaan tuota määrittelemätöntä totuusarvoa. Silloin *kaava1* $\equiv$ *kaava2* pätee automaattisesti aina kun *kaava1* $\Leftrightarrow$ *kaava2* pätee. Silloin edellä olleita yhtätotuuden päättelysääntöjä saa käyttää niin että myös jos-osassa symbolin "$\equiv$" tilalla on "$\Leftrightarrow$". (Yhtäsuuruuden päättelysäännöt eivät tarvitse samankaltaista huomautusta, koska "$\equiv$" ei esiinny niiden jos-osissa.)

On myös olemassa päättelysääntöjä, joiden jos-osassa on alun perinkin "$\Leftrightarrow$" eikä "$\equiv$". Niihin liittyviä ilmiöitä on ehkä helpoin esitellä neliöjuuren avulla. Ilmausten *lauseke1*, *lauseke2* ja *lauseke3* sijaan kirjoitamme $\mathfrak{f}$, $\mathfrak{g}$ ja $\mathfrak{h}$, koska muuten sivusuuntainen tila loppuisi kesken.

Aluksi perustelemme, että $\sqrt{\mathfrak{f}} = \mathfrak{g} \Leftrightarrow \mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2$. Jos $\sqrt{\mathfrak{f}} = \mathfrak{g}$ on tosi, niin $\mathfrak{g} \geq 0$ koska neliöjuuri on aina vähintään 0, ja korottamalla molemmat puolet neliöön nähdään että $\mathfrak{f} = \mathfrak{g}^2$ on tosi. Jos $\mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2$ on tosi, niin $\mathfrak{f} = \mathfrak{g}^2$ takaa että $\mathfrak{g} = \sqrt{\mathfrak{f}} \vee \mathfrak{g} = -\sqrt{\mathfrak{f}}$, ja $\mathfrak{g} \geq 0$ varmistaa, että $\mathfrak{g} = \sqrt{\mathfrak{f}}$ eli $\sqrt{\mathfrak{f}} = \mathfrak{g}$. Tätä yhtäpitävyyttä voidaan käyttää esimerkiksi yhtälöitä ratkaistaessa poistamaan neliöjuuri.

Mutta saattaa olla, että $\sqrt{\mathfrak{f}} = \mathfrak{g}$ tuottaa U ja $\mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2$ tuottaa F. Niin käy esimerkiksi kun sekä $\mathfrak{f}$ että $\mathfrak{g}$ on -1. Silloin $\neg(\sqrt{\mathfrak{f}} = \mathfrak{g}) \Leftrightarrow \neg(\mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2)$ ei päde, koska sen vasen puoli tuottaa U ja oikea puoli tuottaa T. Koska $\neg(\sqrt{\mathfrak{f}} = \mathfrak{g})$ tarkoittaa samaa kuin $\sqrt{\mathfrak{f}} \neq \mathfrak{g}$, seuraa tästä, että neliöjuuren poistaminen tapauksessa $\sqrt{\mathfrak{f}} \neq \mathfrak{g}$ on erilaista kuin tapauksessa $\sqrt{\mathfrak{f}} = \mathfrak{g}$. Mutta tähänkin tapaukseen on kaava, jolla neliöjuuren voi poistaa, sillä $\sqrt{\mathfrak{f}} \neq \mathfrak{g} \Leftrightarrow \mathfrak{f} \geq 0 \wedge (\mathfrak{g} < 0 \vee \mathfrak{f} \neq \mathfrak{g}^2)$ pätee. Sivulla 45 kerrotaan, miten tämän kaavan voi löytää ja miksi se on oikein.

Seuraava sääntö sallii yhtäpitävyyden soveltamisen kaavan sisällä. Sääntö puhuu yhden esiintymän korvaamisesta, mutta useampi esiintymä voidaan korvata korvaamalla yksi kerrallaan.

> **Yhtäpitävyyden käyttö kaavan sisällä:** Jos korvataan yksi *kaava1*:n esiintymä *kaava2*:lla ja jos mikään *kaava1*:n tai *kaava2*:n

vapaa muuttuja, joka esiintyy vapaana myös voimassa olevissa oletuksissa, ei joudu sidotuksi $isompi\_kaava(kaava1)$:ssä eikä $isompi\_kaava(kaava2)$:ssa, niin seuraavissa tapauksissa $isompi\_kaava(kaava1) \Leftrightarrow isompi\_kaava(kaava2)$:

- Jos korvattava esiintymä on parillisen määrän $\neg$:ta alaisuudessa ja $kaava1 \Leftrightarrow kaava2$.
- Jos korvattava esiintymä on parittoman määrän $\neg$:ta alaisuudessa ja $\neg kaava1 \Leftrightarrow \neg kaava2$.

Perustelemme seuraavaksi, että tämä päättelysääntö on pätevä. Käytämme ilmausta ”*kaava* tuottaa enintään U” tarkoittamaan, että tarkasteltavassa tilanteessa *kaava* tuottaa F tai U (siis se ei tuota T). Nyt $kaava1 \Leftrightarrow kaava2$ tarkoittaa samaa kuin että jokaisessa mahdollisessa tilanteessa $kaava1$ ja $kaava2$ tuottavat T tai molemmat tuottavat enintään U.

Vastaavasti ”*kaava* tuottaa vähintään U” tarkoittaa, että se tuottaa U tai T (siis se ei tuota F). Otamme käyttöön tilapäisen merkinnän $P \Updownarrow Q$ tarkoittamaan $\neg P \Leftrightarrow \neg Q$. Siis $kaava1 \Updownarrow kaava2$ tarkoittaa samaa kuin että jokaisessa mahdollisessa tilanteessa $kaava1$ ja $kaava2$ tuottavat F tai molemmat tuottavat vähintään U. Parittoman tapauksen lähtökohta $\neg kaava1 \Leftrightarrow \neg kaava2$ voidaan kirjoittaa myös muodossa $kaava1 \Updownarrow kaava2$.

Olkoot ik, K_1 ja K_2 kuten sivulla 40.

Oletetaan $K_1 \Leftrightarrow K_2$. Tarkastellaan mitä tahansa mahdollista tilannetta.

Jos K_1 tuottaa T, niin myös K_2 tuottaa T. Kuvan 3 mukaan myös $K_1 \vee muu$, $K_2 \vee muu$, $muu \vee K_1$ ja $muu \vee K_2$ tuottavat T. Samoin käy jos muu tuottaa T. Muussa tapauksessa K_1, K_2, muu, $K_1 \vee muu$, $K_2 \vee muu$, $muu \vee K_1$ ja $muu \vee K_2$ tuottavat enintään U. Siksi $K_1 \vee muu \Leftrightarrow K_2 \vee muu$ ja $muu \vee K_1 \Leftrightarrow muu \vee K_2$.

Jos K_1 tuottaa enintään U, niin myös K_2 tuottaa enintään U. Tällöin myös $K_1 \wedge muu$, $K_2 \wedge muu$, $muu \wedge K_1$ ja $muu \wedge K_2$ tuottavat enintään U. Samoin käy jos muu tuottaa enintään U. Muussa tapauksessa K_1, K_2, muu, $K_1 \wedge muu$, $K_2 \wedge muu$, $muu \wedge K_1$ ja $muu \wedge K_2$ tuottavat T. Siksi $K_1 \wedge muu \Leftrightarrow K_2 \wedge muu$ ja $muu \wedge K_1 \Leftrightarrow muu \wedge K_2$.

Koska $\neg\neg K_1 \equiv K_1$ ja $K_2 \equiv \neg\neg K_2$, pätee $\neg\neg K_1 \Leftrightarrow \neg\neg K_2$ eli $\neg K_1 \Updownarrow \neg K_2$.

Olkoon x muuttuja, joka ei esiinny vapaana voimassa olevissa oletuksissa. Tällöin tilanne on mahdollinen x:n arvosta riippumatta. Jos jollakin x:n arvolla K_1 tuottaa T, niin samalla x:n arvolla K_2 tuottaa T. Tällöin $\exists x : K_1$ ja $\exists x : K_2$ tuottavat T. Muussa tapauksessa kullakin x:n arvolla K_1 ja K_2 tuottavat enintään U. Tällöin $\exists x : K_1$ ja $\exists x : K_2$ tuottavat enintään U. Siksi $\exists x : K_1 \Leftrightarrow \exists x : K_2$. Jos jokaisella x:n arvolla K_1 tuottaa T, niin jokaisella x:n arvolla K_2 tuottaa T. Tällöin $\forall x : K_1$ ja $\forall x : K_2$ tuottavat T. Muussa tapauksessa jollakin x:n arvolla K_1 ja K_2 tuottavat enintään U. Tällöin $\forall x : K_1$ ja $\forall x : K_2$ tuottavat enintään U. Siksi $\forall x : K_1 \Leftrightarrow \forall x : K_2$.

Samalla tavalla voidaan osoittaa, että jos $K_1 \Updownarrow K_2$, niin $K_1 \vee muu \Updownarrow K_2 \vee muu$, $\neg K_1 \Leftrightarrow \neg K_2$, $\exists x : K_1 \Updownarrow \exists x : K_2$ ja niin edelleen.

Näistä seuraa samalla tavalla kuin sivun 40 todistuksessa, että jos parillisessa tapauksessa $kaava1 \Leftrightarrow kaava2$ tai parittomassa tapauksessa $kaava1 \Updownarrow kaava2$, niin $isompi\_kaava(kaava1) \Leftrightarrow isompi\_kaava(kaava2)$. Todistus on valmis.

Yksi usein hyvä keino käsitellä määrittelemättömiä toimintoja on poistaa ne päättelyn alkuvaiheissa, jonka jälkeen symbolien ”$\equiv$” ja ”$\Leftrightarrow$” erosta ei tarvitse välittää. Seuraavaksi näytämme, miten poistaminen onnistuu juuri käsittelemällämme päättelysäännöllä.

Määrittelemättömän toiminnon poistamiseksi tällä päättelysäännöllä tarvitaan kaksi kaavaa: yksi, joka kertoo milloin toiminto on määritelty, ja toinen, joka esittää toiminnon vaikutuksen toiminnoilla, jotka ovat aina määritellyt — tai ainakin useammin määritellyt kuin alkuperäinen kaava. Kutsumme näitä nimillä *määr* ja *vaik*, ja alkuperäistä kaavaa nimellä *alkup*. Esimerkiksi kun *alkup* on $\sqrt{\mathfrak{f}} = \mathfrak{g}$ voidaan valita *määr*:ksi $\mathfrak{f} \geq 0$ ja *vaik*:ksi $\mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2$. Kaavan *määr* täytyy olla aina määritelty.

Jos *alkup* on parillisen määrän $\neg$:ta alaisuudessa, niin aloitetaan yhtäpitävyydestä $alkup \Leftrightarrow määr \wedge vaik$. Se on yhtäpitävyys, koska niissä tilanteissa joissa *alkup* ei ole määritelty tuottaa vasen puoli U ja oikea puoli F, ja muissa tilanteissa *määr* tuottaa T ja *alkup* sanoo saman kuin *vaik*. Yhtäpitävyyden käyttö kaavan sisällä tuottaa $isompi\_kaava(alkup) \Leftrightarrow isompi\_kaava(määr \wedge vaik)$. Neliöjuuriesimerkissämme $määr \wedge vaik \equiv \mathfrak{f} \geq 0 \wedge \mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2 \equiv \mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2$, koska $\mathfrak{f} = \mathfrak{g}^2$ takaa $\mathfrak{f} \geq 0$. Siksi $isompi\_kaava(\sqrt{\mathfrak{f}} = \mathfrak{g}) \Leftrightarrow isompi\_kaava(\mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2)$ parillisessa tapauksessa. Täällä on muutama neliöjuuriyhtälö!

Jos *alkup* on parittoman määrän $\neg$:ta alaisuudessa, niin $isompi\_kaava(alkup) \Leftrightarrow isompi\_kaava(\neg määr \vee vaik)$. Nähdäksemme, että näin on, aloitamme siitä, että kun *alkup* ei ole määritelty, $\neg alkup$ ja $määr \wedge \neg vaik$ tuottavat U ja F, ja muulloin ne tarkoittavat samaa. Siksi $\neg alkup \Leftrightarrow määr \wedge \neg vaik \equiv \neg(\neg määr \vee vaik)$, joten yhtäpitävyyden käyttösäännön ehto toteutuu. Neliöjuuriesimerkissämme $\neg määr \vee vaik$ on $\mathfrak{f} < 0 \vee \mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2$, joten $isompi\_kaava(\sqrt{\mathfrak{f}} = \mathfrak{g}) \Leftrightarrow isompi\_kaava(\mathfrak{f} < 0 \vee \mathfrak{g} \geq 0 \wedge \mathfrak{f} = \mathfrak{g}^2)$. Jos $isompi\_kaava(X)$ on $\neg X$, niin saadaan $\sqrt{\mathfrak{f}} \neq \mathfrak{g} \Leftrightarrow \mathfrak{f} \geq 0 \wedge (\mathfrak{g} < 0 \vee \mathfrak{f} \neq \mathfrak{g}^2)$, eli juuri se yhtäpitävyys, jota ihmeteltiin sivulla 43.

Tässä on vielä kaksi esimerkkiä parillisista tapauksista. Jälkimmäisessä on otettu huomioon, että vertailun suunta kääntyy, kun sen molemmat puolet kerrotaan negatiivisella luvulla.

$$isompi\_kaava(\tfrac{\mathfrak{f}}{\mathfrak{g}} = \mathfrak{h}) \quad \Leftrightarrow \quad isompi\_kaava(\mathfrak{g} \neq 0 \wedge \mathfrak{f} = \mathfrak{g}\mathfrak{h})$$
$$isompi\_kaava(\tfrac{\mathfrak{f}}{\mathfrak{g}} \leq \mathfrak{h}) \quad \Leftrightarrow \quad isompi\_kaava(\mathfrak{g} < 0 \wedge \mathfrak{f} \geq \mathfrak{g}\mathfrak{h} \vee \mathfrak{g} > 0 \wedge \mathfrak{f} \leq \mathfrak{g}\mathfrak{h})$$

Samat parittomina tapauksina:

$$\mathit{isompi\_kaava}(\tfrac{\mathfrak{f}}{\mathfrak{g}} = \mathfrak{h}) \;\Leftrightarrow\; \mathit{isompi\_kaava}(\mathfrak{g} = 0 \vee \mathfrak{f} = \mathfrak{gh})$$
$$\mathit{isompi\_kaava}(\tfrac{\mathfrak{f}}{\mathfrak{g}} \leq \mathfrak{h}) \;\Leftrightarrow\; \mathit{isompi\_kaava}(\mathfrak{g} = 0 \vee \mathfrak{g} < 0 \wedge \mathfrak{f} \geq \mathfrak{gh} \vee \mathfrak{g} > 0 \wedge \mathfrak{f} \leq \mathfrak{gh})$$

Niin kauan kun määrittelemättömiä toimintoja on mukana, joudutaan varmistamaan, että ne otetaan oikealla tavalla huomioon. Tämän luvun lopuksi käsitellään paria keinoa, jotka helpottavat tätä.

Laskutoimitus tai muu sellainen on *tiukka* (*strict*), jos ja vain jos aina kun ainakin yksi lähtökohta on määrittelemätön, myös tulos on määrittelemätön. Koulumatematiikan laskutoimitukset ovat tiukkoja. Esimerkiksi $0 \cdot \frac{1}{x}$ ei ole aina nolla, vaan se on määrittelemätön kun $x = 0$, koska $\frac{1}{0}$ on määrittelemätön ja kertolasku on tiukka.

Tarkastelemme mitä tahansa lakia muotoa $\mathit{lauseke}1 = \mathit{lauseke}2$, jonka sisältämät kaikki laskutoimitukset ovat tiukkoja. Kun kaavan sisällä käytetään tätä lakia siten, että sen muuttujien paikalle sijoitetaan lausekkeet, niin määrittelemättömyys tarvitsee ottaa huomioon vain jos lain jommallakummalla puolella esiintyy muuttuja, joka ei esiinny vastakkaisella puolella. Esimerkiksi lain $0a = 0$ vasemmalla puolella esiintyy muuttuja a, joka ei esiinny oikealla puolella. Siksi $0a = 0$ ei oikeuta päättelemään $x + 0 \cdot \frac{1}{x-5} = 5 \Leftrightarrow x + 0 = 5 \Leftrightarrow x = 5$.

Mutta lain $a + b = b + a$ molemmilla puolilla on täsmälleen samat muuttujat a ja b, joten se oikeuttaa päättelemään $\mathit{isompi\_kaava}(\mathit{lauseke}1 + \mathit{lauseke}2) \Leftrightarrow \mathit{isompi\_kaava}(\mathit{lauseke}2 + \mathit{lauseke}1)$ vaikka $\mathit{lauseke}1$ tai $\mathit{lauseke}2$ tai molemmat olisivat määrittelemättömiä. Tämä johtuu siitä, että aina kun $\mathit{lauseke}1 + \mathit{lauseke}2$ on määrittelemätön, on myös $\mathit{lauseke}2 + \mathit{lauseke}1$ määrittelemätön. Ne ovat siis silloinkin laajennetusti yhtäsuuret.

Symboleita ”$\wedge$”, ”$\vee$” ja ”$\neg$” koskevissa päättelyissä voidaan hyödyntää seuraavaa tosiasiaa:

> Jos $\mathit{kaava}1 \Leftrightarrow \mathit{kaava}2$, jos $\mathit{kaava}1$ ja $\mathit{kaava}2$ eivät sisällä muita symboleita kuin ”F”, ”T”, ”$\wedge$”, ”$\vee$”, ”$\neg$”, ”(”, ”)” ja propositiomuuttujia, ja jos mikään propositiomuuttuja ei ole $\mathit{kaava}1$:ssä sekä parillisen että parittoman määrän $\neg$:ta alaisuudessa ja samoin $\mathit{kaava}2$:lle, niin $\mathit{kaava}1 \equiv \mathit{kaava}2$.

Jätämme tämän tosiasian perustelematta, koska perustelu vaatisi syvälle kolmiarvologiikkaan menemistä. (Se käyttää kolmiarvologiikan käsitettä ”regularity”.)

Oppikirjoissa usein esitettävistä yhtäpitävyyslaeista vain neljä käyttää pelkästään edellä mainittuja symboleita, mutta rikkoo parillisuuden ja parittomuuden ehtoa. Millekään niistä $\mathit{kaava}1 \equiv \mathit{kaava}2$ ei päde. Ne ovat:

$$
\begin{array}{rcl}
P \vee \neg P & \Leftrightarrow & \mathsf{T} \\
P \wedge \neg P & \Leftrightarrow & \mathsf{F} \\
P \wedge (\neg P \vee Q) & \Leftrightarrow & P \wedge Q \\
P \vee (\neg P \wedge Q) & \Leftrightarrow & P \vee Q
\end{array}
$$

Kahdella jälkimmäisellä on mielenkiintoinen yhteys ohjelmointiin. Sitä voi havainnollistaa esimerkillä seikkailupelistä. Pelaajan edessä on ovi. Sen takana saattaa olla aarre. Mutta oveen saattaa olla kytketty pommi, joka räjähtää, jos ovi avataan. Jos oven avaus johtaa räjähdykseen, niin peli päättyy pelaajan häviöön eikä pelaaja saa tietää, oliko oven takana aarretta. Muussa tapauksessa peli jatkuu joko aarteen kanssa tai ilman. Pelaaja on matkansa varrella onnistunut hankkimaan pommi-ilmaisimen.

Monissa nykyaikaisissa ohjelmointikielissä ilmaus ”¬pommi ja avaus” toimii siten, että ensin tutkitaan, onko ovessa pommia. Ovi avataan vain jos pommia ei ole. Vastaavasti ”avaus ja ¬pommi” toimii siten, että ensin avataan ovi. Pommi-ilmaisinta käytetään vain jos ovi aukeni turvallisesti ja sen takana oli aarre. Jos U tulkitaan tarkoittamaan räjähdystä, niin mahdollisuudet ovat seuraavat:

pommi	avaus	¬pommi ja avaus	avaus ja ¬pommi
F	F	F	F
F	T	T	T
T	U	F	U

Tällainen käyttäytyminen ei vastaa kolmiarvologiikan kaavaa $avaus \wedge \neg pommi$, mutta se vastaa kolmiarvologiikan kaavaa $avaus \wedge (\neg avaus \vee \neg pommi)$.

Jokainen luvuissa 3.1 ja 3.3 esitetty kvanttorien laki muotoa $kaava1 \Leftrightarrow kaava2$ on pätevä myös muodossa $kaava1 \equiv kaava2$, ellei lain kohdalla ole sanottu toisin.

Tarkastamme esimerkin vuoksi lain ”jos x ei esiinny vapaana $kaava1$:ssä, niin $\exists x : \big(kaava1 \wedge kaava2(x)\big) \equiv kaava1 \wedge \exists x : kaava2(x)$”. Sivulla 33 huomasimme, että jos tarkasteltavassa tilanteessa $kaava1$ tuottaa F, niin kumpikin puoli sievenee nopeasti muotoon F; ja jos $kaava1$ tuottaa T, niin kumpikin puoli sievenee heti muotoon $\exists x : kaava2(x)$. Jäljellä ovat enää tilanteet, joissa $kaava1$ tuottaa U. Jos jokaisella x:n arvolla $kaava2(x)$ tuottaa F, niin kumpikin puoli tuottaa F. Muussa tapauksessa jollakin x:n arvolla $kaava2(x)$ tuottaa U tai T. Pätee $\mathsf{U} \wedge \mathsf{U} \equiv \mathsf{U} \wedge \mathsf{T} \equiv \mathsf{U}$. Siksi $\exists x : kaava2(x)$ tuottaa U tai T, vasen puoli tuottaa U ja oikea puoli tuottaa U.

3.3 Kaikkikvanttori

Symboli ”$\forall$” on *kaikkikvanttori* (*universal quantifier*). Ilmaus $\forall x : kaava(x)$ väittää, että jos x:lle valitaan mikä tahansa arvo, niin $kaava(x)$ on tosi.

Kaikkikvanttorin opiskelemista vaikeuttaa se, että matematiikassa on hyvin tavallista jättää se kirjoittamatta. Esimerkiksi kun halutaan sanoa, että luku ei muutu kun sen kertoo ykkösellä, niin usein kirjoitetaan $1a = a$ eikä $\forall a : 1a = a$. Tämä vaikeuttaa ilmausten $kaava(x)$ ja $\forall x : kaava(x)$ välisen eron hoksaamista.

Kaava $\forall x : kaava(x)$ väittää jotakin jokaisesta puheenaiheen arvosta. Ei ole olennaista, että niistä arvoista puhuvan muuttujan nimi on x (sivu 28). Jos esimerkiksi a ei esiinny $kaava(x)$:ssä, niin $\forall a : kaava(a)$ sanoo saman kuin $\forall x : kaava(x)$. Mutta $kaava(x)$ väittää jotakin vain siitä tai niistä arvoista, joiden sillä hetkellä ajatellaan voivan olla siinä muuttujassa, jonka nimi on nimenomaan x. Kokeile tätä eroa tekemällä tämä tehtävä.

Riippuu asiayhteydestä, mitä arvoja x:ssä voi olla. Kun sanotaan ”ratkaise yhtälö $5x - 7 = 13$”, niin x:ssä voi olla vain sellaisia arvoja, että $5x - 7 = 13$ on tosi. Jos sanotaan ”tarkastelemme ensin tapausta $x < 0$”, niin siitä alkaen x voi saada vain negatiivisia arvoja kunnes tapaus on valmis. Jos heti sen jälkeen sanotaan ”muussa tapauksessa”, niin siitä alkaen x voi olla vain nolla tai positiivinen kunnes tapaus on valmis.

Jollei x:stä ole sanottu mitään, niin siinä voi olla mikä tahansa puheenaiheen arvo. Silloin $kaava(x)$ tuottaa saman totuusarvon kuin $\forall x : kaava(x)$. Silloinkin ne ovat sikäli eri kaavat, että jos ne eivät ole tosi, niin $kaava(x)$:n vastaesimerkissä annetaan x:lle arvo, mutta $(\forall x : kaava(x))$:n vastaesimerkissä ei anneta.

Asiayhteys ei samalla tavalla vaikuta kvanttorilla luodun muuttujan arvoihin, koska kvanttorilla luotu muuttuja on eri muuttuja kuin kvanttorin vaikutusalueen ulkopuolella sijaitseva, vaikka niillä olisi sama nimi. Kuitenkin toisinaan olisi kätevää rajoittaa myös kvanttorilla luodulle muuttujalle mahdollisia arvoja. Sitä varten MathCheckissä voi asettaa rajoitteen tyyliin $\forall x$; $rajoite(x) : kaava(x)$ (toimii myös ”$\exists$”:lle).

Tähän aihepiiriin liittyy tämä tehtävä.

Yritä tehdä vielä tämä tehtäväryhmä, mutta ei haittaa, jollet onnistu. Sitten jatka lukemista.

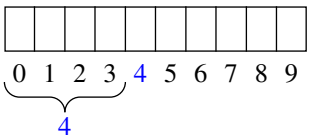
Tehtävän ensimmäisessä vaiheessa pitää keksiä kaava $P(x)$, joka alkaa ”$2x \geq$” ja (jotta $P(x) \Leftrightarrow \mathsf{T}$ pätisi) joka on aina tosi. Esimerkiksi $2x \geq 2x$ on sellainen. Koska se on aina tosi, se toteuttaa päättelyn $\forall x : P(x) \Leftrightarrow P(x) \Leftrightarrow \mathsf{T}$ molemmat yhtäpitävyydet eli kelpaa vastaukseksi. Myös $2x \geq 1 \vee \mathsf{T}$ alkaa ”$2x \geq$” ja toteuttaa päättelyn. Se ei oikeastaan ole tehtävän hengen mukainen, mutta kelpaa silti vastaukseksi.

Sitten pitää keksiä samoin alkava kaava $P(x)$, joka ei saa olla aina tosi, jotta $P(x) \Leftrightarrow \mathsf{T}$ ei pätisi. Koska se on jollakin x:n arvolla epätosi, on $\forall x : P(x)$ epätosi. Jotta $\forall x : P(x) \Leftrightarrow P(x)$ pätisi, pitää $P(x)$:n olla x:n arvosta riippumatta epätosi. Toisaalta mikä tahansa sellainen kaava toteuttaa ensimmäisen yhtäpitävyyden mutta ei jälkimmäistä, ja niin ollen kelpaa vastaukseksi. Esimerkkejä ovat $2x \geq 2x + 1$ ja $2x \geq 1 \wedge \mathsf{F}$.

Lopuksi pitää keksiä kaava $P(x)$, jolle $\forall x : P(x) \Leftrightarrow P(x)$ ei päde. Kahden ensimmäisen kohdan vuoksi tiedämme, että sen pitää olla tosi jollakin x:n arvolla mutta ei kaikilla. Toisaalta jokainen sellainen kelpaa, sillä $\forall x : P(x)$ on sille epätosi, joten $\forall x : P(x) \Leftrightarrow P(x)$ ei päde kun $P(x)$ on tosi. Esimerkiksi $2x \geq 1$ kelpaa.

Nyt otamme puheenaiheeksi luvuista koostuvat taulukot, sillä niistä saa monipuolisia automaattisesti tarkastettavissa olevia tehtäviä tyyppiä ”kirjoita kaava, joka sanoo, että ...”. Esimerkiksi $A[1 \ldots 4]$ koostuu neljästä luvusta, joita merkitään $A[1]$, $A[2]$, $A[3]$ ja $A[4]$. Kaava $\exists i \,;\, 1 \leq i \leq 4 : A[i] = 6$ sanoo, että jokin näistä luvuista on 6. Se tarkoittaa samaa kuin $A[1] = 6 \vee A[2] = 6 \vee A[3] = 6 \vee A[4] = 6$. Se on tosi esimerkiksi kun A on $[3,2,6,2]$, mutta ei ole tosi kun A on $[3,8,1,0]$.

Hakasulkeiden ”[” ja ”]” välissä oleva luku valitsee, mistä taulukon alkiosta puhutaan. Sitä sanotaan taulukon *indeksiksi*. Taulukon nimi, pienin sallittu indeksi ja suurin sallittu indeksi voidaan ilmoittaa lyhyesti muodossa *nimi*[*pienin* ... *suurin*]. Pienin indeksi on hyvin usein joko 0 tai 1. Ykkönen on yleinen arkimaailmassa ja matematiikassa, koska sitä käytettäessä ”ensimmäinen”, ”toinen”, ”kolmas”, ... täsmäävät indeksiin, eli esimerkiksi kolmas alkio on $A[3]$. Nolla on yleinen ohjelmoinnissa ja osittain matematiikassa, koska niissä on etua siitä, että kuten alla oleva kuva havainnollistaa, $A[i]$ sijaitsee i alkion verran eteenpäin A:n alusta.



Koska taulukoiden koot vaihtelevat, ilmaistaan suurin sallittu indeksi usein muuttujan avulla. Esimerkiksi sekä $A[1 \ldots n]$ että $A[0 \ldots n-1]$ sisältävät n alkiota. Taulukko voi olla myös tyhjä eli sisältää 0 alkiota. Sitä varten ilmauksessa $A[a \ldots y]$ sallitaan $y = a - 1$, eli esimerkiksi $A[1 \ldots 0]$ ja $A[0 \ldots -1]$ sallitaan. Sen sijaan $A[1 \ldots -1]$ ja $A[5 \ldots 3]$ on kielletty, koska ne väittävät, että A:ssa on miinus yksi alkiota. Harjoittele näitä asioita tekemällä tämä tehtäväryhmä!

Sen, että taulukon $A[1 \ldots n]$ alkiot ovat kasvavassa järjestyksessä, voi ilmaista kaavalla $\forall i \,;\, 1 \leq i < n : A[i] \leq A[i+1]$. Se sanoo, että seuraava alkio on aina vähintään yhtäsuuri kuin edellinen alkio. Kokeile sitä täällä. Sitten ennusta, minkälainen palaute samasta vastauslaatikosta tulee kaavalle $\forall i \,;\, 1 \leq i < n : A[i] < A[i+1]$, ja kokeile osuiko ennustuksesi oikeaan.

Kokeiltava kaava vaatii enemmän kuin edellinen: se vaatii myös, että mitkään kaksi peräkkäistä alkiota eivät ole yhtäsuuret. Siksi MathCheck hylkää sen ja antaa vastaesimerkiksi jonkin taulukon, joka on kasvavassa järjestyksessä, mutta jossa on kaksi yhtäsuurta alkiota peräkkäin. Esimerkiksi $[0,0]$ on sellainen. Lisäksi

MathCheck kertoo sen muuttujan arvon, jonka avulla suurin sallittu indeksi määrättiin. Tässä tapauksessa se muuttuja on n. MathCheck kertoo myös, että piilossa oleva mallivastaus on tällä vastaesimerkillä tosi, mutta vastauslaatikkoon kirjoitettu vastaus on epätosi.

Sitten ennusta, minkälainen palaute samasta vastauslaatikosta tulee kaavalle $\forall i\,;\, 1 < i < n : A[i] \leq A[i+1]$, ja kokeile. Jos palaute ei vastaa ennustettasi, niin yritä ymmärtää se ennen kuin luet eteenpäin.

Tämä kaava on muuten oikein, mutta ei ota huomioon taulukon ensimmäistä alkiota eli $A[1]$:tä. Siksi MathCheck antaa vastaesimerkiksi taulukon, jonka ensimmäinen alkio on suurempi kuin toinen, mutta joka on toisesta alkiosta alkaen järjestyksessä. Sille mallivastaus on epätosi, mutta vastauslaatikon kaava on tosi.

Seuraavaksi on vuorossa $\forall i\,;\, 1 \leq i \leq n : A[i] \leq A[i+1]$. Nytkin, ja koko ajan jatkossa, jos palaute ei vastaa ennustettasi, niin yritä ymmärtää se ennen kuin luet eteenpäin.

Jos taulukko on epätyhjä, niin tämä kaava vaatii sen olevan järjestyksessä alkiosta $A[1]$ alkioon $A[n+1]$ saakka. Suurin i:n saama arvo on n, ja sillä kaava kokeilee päteekö $A[n] \leq A[n+1]$. Taulukossa ei kuitenkaan ole alkiota $A[n+1]$, joten vertailua, onko se vähintään yhtäsuuri kuin $A[n]$, ei voida suorittaa. Ohjelmoinnissa tällainen tilanne voi aiheuttaa toimintavirheen. MathCheck reagoi siihen antamalla vastaesimerkin, jolla kaava ei ole tosi eikä epätosi, vaan sen totuusarvo on U eli määrittelemätön (undefined). Kun teet myöhempiä taulukkotehtäviä, niin saatat saada palautteessa U:n. Todennäköinen (mutta ei ainoa mahdollinen) syy on, että kaavasi indeksoi taulukkoa ohi sen reunan.

Taulukon voi väittää olevan kasvavassa järjestyksessä myös kaavalla, joka ei vertaa kutakin alkiota seuraavaan, vaan vertaa jokaista alkiota jokaiseen myöhemmin tulevaan alkioon. Esimerkiksi $\forall i : \forall j\,;\, 1 \leq i < j \leq n : A[i] \leq A[j]$ on sellainen. Voit kokeilla myös sitä! Mitä luulet, toimiiko myös $\forall i : \forall j\,;\, 1 \leq i \leq j \leq n : A[i] \leq A[j]$? Ennusta, ja sitten kokeile!

Tehtäväryhmässä kokeillaan seuraavaksi erilaisia vastauksia, kun mallivastaus on $\forall i\,;\, 1 \leq i < n : A[i] < A[i+1]$. Se sanoo, että A on *aidosti kasvavassa* järjestyksessä. Se tarkoittaa, että jokainen alkio (paitsi ensimmäinen) on suurempi kuin edellinen alkio. Jo tutuksi tullut kaava $\forall i\,;\, 1 \leq i < n : A[i] \leq A[i+1]$ sanoo, että A on kasvavassa järjestyksessä. Ennusta sille palaute ja kokeile.

Palautteeksi tuli taulukko, joka on kasvavassa mutta ei aidosti kasvavassa järjestyksessä. Sille mallivastaus on epätosi mutta vastauslaatikon kaava tosi.

Seuraavaksi ennusta ja kokeile kaavaa $\forall i : \forall j\,;\, 1 \leq i < j \leq n : A[i] < A[j]$, ja lopuksi kaavaa $\forall i : \forall j\,;\, 1 \leq i \leq j \leq n : A[i] < A[j]$.

Näistä kahdesta ensimmäinen kelpasi MathCheckille, sillä se todellakin sanoo, että A on aidosti kasvavassa järjestyksessä. Sen sijaan jälkimmäinen kaava sai yhden alkion kokoisen vastaesimerkin. Koska kaavassa lukee $i \leq j$, niin se vaatii kullakin i:n arvolla $A[i] < A[i]$. Tyhjälle taulukolle tämä ei aiheuta mitään, koska

jos $n = 0$, niin $1 \leq i \leq j \leq n$ ei toteudu millään i ja j. Siksi kaava on tosi, kun $n = 0$. Mutta epätyhjälle taulukolle $1 \leq i \leq j \leq n$ on tosi ainakin kun $i = j = 1$, joten kaava vaatii ainakin $A[1] < A[1]$. Se ei ole tosi. Siksi kaava on epätosi, kun $n \geq 1$. Niinpä kaava on vain monimutkainen tapa sanoa, että A on tyhjä. Tehtäväryhmän viimeisessä vastauslaatikossa voit kokeilla, että näin todella on.

Näimme, että $\forall i\,;\, 1 \leq i < n : A[i] \bowtie A[i+1]$ tarkoittaa samaa kuin $\forall i : \forall j\,;\, 1 \leq i < j \leq n : A[i] \bowtie A[j]$, kun "$\bowtie$" on "$\leq$" tai "$<$". Tarkoittavatko ne samaa, kun "$\bowtie$" on "$=$"? Entä kun "$\bowtie$" on "$\neq$"? Ennusta ja kokeile!

Ovatko yksialkioiset taulukot kasvavassa järjestyksessä? Entä tyhjä taulukko? Logiikan kanta on, että ovat. Jotta taulukko olisi epäjärjestyksessä, siinä täytyy olla jokin kohta, jossa oleva alkio on suurempi kuin seuraava alkio. Sellaista kohtaa ei voi olla, jos alkioita on vähemmän kuin kaksi.

Jos taulukossa A on täsmälleen yksi alkio, niin $n = 1$. Silloin $1 \leq i \leq j \leq n$ tarkoittaa samaa kuin $1 \leq i \leq j \leq 1$. Se toteutuu täsmälleen silloin kun $i = j = 1$. Niinpä $\forall i : \forall j\,;\, 1 \leq i \leq j \leq 1 : A[i] \leq A[j]$ tarkoittaa samaa kuin $A[1] \leq A[1]$. Koska se on tosi, on $\forall i : \forall j\,;\, 1 \leq i \leq j \leq n : A[i] \leq A[j]$ yhden alkion taulukoilla tosi. Niin sen kuuluu ollakin, koska jokainen yhden alkion taulukko on järjestyksessä.

Sekä $\forall i : \forall j\,;\, 1 \leq i < j \leq n : A[i] \leq A[j]$ että $\forall i : \forall j\,;\, 1 \leq i \leq j \leq n : A[i] \leq A[j]$ kelpasivat edellä sanomaan, että A on kasvavassa järjestyksessä. Niinpä myös $\forall i : \forall j\,;\, 1 \leq i < j \leq n : A[i] \leq A[j]$ on tosi, kun $n = 1$. Mutta silloin $1 \leq i < j \leq n$ on epätosi: sen mukaan i:n on oltava vähintään 1, j saa olla enintään 1, mutta silti i:n pitää olla pienempi kuin j. Siis i:llä ja j:llä ei voi olla arvoja, mutta silti koko kaava on tosi! Yksi tapa ymmärtää tämä on, että kaava tekee n:n arvosta riippuen nolla tai useampia testejä, päteekö $A[i] \leq A[j]$. Esimerkiksi kun $n = 3$, se testaa pätevätkö $A[1] \leq A[2]$, $A[1] \leq A[3]$ ja $A[2] \leq A[3]$. Kaava on tosi jos ja vain jos jokainen testi menee läpi. Arvolla $n = 1$ ei tehdä yhtään testiä, joten silloin jokainen testi menee läpi.

Myöskään $\forall i\,;\, 1 \leq i < n : A[i] \leq A[i+1]$ ei tee yhtään suuruusjärjestystestiä, kun $n = 1$. Kun $n = 0$, ei mikään oikein toimiva kaava voi tehdä yhtään suuruusjärjestystestiä, koska ei ole yhtään alkiota, jota voisi verrata mihinkään alkioon.

Siis kun $rajoite(x)$ ei salli x:lle yhtään arvoa, on $\forall x\,;\, rajoite(x) : kaava(x)$ tosi riippumatta siitä, mitä $kaava(x)$ sanoo. Tämä on esimerkki laajasti vallitsevasta periaatteesta, joka näyttää olevan yksi vaikeimmista logiikan opiskelussa: *mahdottomasta seuraa mitä tahansa*. Englanniksi sitä kutsutaan räväkällä ilmauksella *principle of explosion*. Se näyttää tuottavan niin hulluja lopputuloksia, että niitä ei voi hyväksyä — mutta se ei tuota niitä oikeasti, vaan vain näennäisesti. Esimerkiksi kaava $\forall x\,;\, 2 \leq x \leq 1 : 3 = 4$ on todellakin tosi, vaikka se näyttää väittävän, että $3 = 4$. Mutta se ei todellisuudessa väitä, että $3 = 4$, koska se lupaa $3 = 4$ vain ehdolla $2 \leq x \leq 1$, joka ei koskaan ole tosi.

Kokeile täällä, mitä mieltä MathCheck on kaavasta $\forall x\,;\, 2 \leq x \leq 1 : 3 = 4$. Sitten

vaihda ykkösen tilalle 2, ennusta mitä palautteeksi tulee ja kokeile, ennustitko oikein. Hieman alempana on vastauslaatikko samaa varten kaavalle $\forall x\,;\, 2 \leq x \leq 1 : 4 = 4$, ja sitä alempana kaavoille, joissa kaikkikvanttorin tilalla on olemassaolokvanttori.

Joskus kaavan laatiminen menee pieleen sellaisella tavalla, että kaava sanoo jotain aivan muuta kuin oli tarkoitus. Esimerkiksi jos yritetään sanoa, että A on kasvavassa järjestyksessä eikä mikään alkio toistu, niin ei tarvita kuin pieni lipsahdus, niin kaavaksi voi tulla $\forall i : \forall j\,;\, 1 \leq i \leq j \leq n : A[i] < A[j]$. Kuten edellä näimme, se sanoo, että A on tyhjä.

Nyt on aika harjoitella itsenäisesti taulukoista puhuvien kaavojen kirjoittamista. Neljä tehtävää! Toiset neljä!

Arkijärki sanoo, että jos ei ole niin, että jokainen koira on ruskea, niin on olemassa koira, joka ei ole ruskea. Arkijärki sanoo saman myös toisinpäin: jos on olemassa koira, joka ei ole ruskea, niin ei ole niin, että jokainen koira on ruskea. Arkijärjen mukaan tämä ei koske pelkästään koiria ja ruskeutta, vaan aiheutuu sanojen ”ei”, ”jokainen” ja ”olemassa” merkityksistä. Vaikka emme tietäisi mitä ”vinksis” ja ”töksis” tarkoittavat, niin silti vaikuttaa todelta, että jos ei ole niin, että jokainen vinksis on töksis, niin on olemassa vinksis joka ei ole töksis, ja toisinpäin.

Siis $\neg\forall x : kaava(x) \Leftrightarrow \exists x : \neg kaava(x)$ on logiikan laki. Luvussa 5 esitettävin keinoin sille voi laatia paljon matemaattisemman todistuksen kuin ”vinksis” ja ”töksis”. Mutta jos ”vinksis” ja ”töksis” auttavat muistamaan sen ja että se on laki, niin ne ovat tehneet tehtävänsä.

Myös $\neg\exists x : kaava(x) \Leftrightarrow \forall x : \neg kaava(x)$ on logiikan laki. Sen voi johtaa edellisestä, johtaa muulla tavalla tai perustella vinksiksen ja töksiksen avulla.

Johtaaksemme sen edellisestä, perustelemme ensin, että myös $\neg\forall x : kaava(x) \equiv \exists x : \neg kaava(x)$ on logiikan laki. Perustelu nojautuu ”¬”:sta kuvassa 3 ja kvanttoreista sivulla 35 kerrottuun. Niiden mukaan $\neg\forall x : kaava(x)$ tuottaa U jos ja vain jos $\forall x : kaava(x)$ tuottaa U jos ja vain jos $kaava(x)$ tuottaa U jollakin x:n arvolla eikä tuota F millään x:n arvolla jos ja vain jos $\exists x : \neg kaava(x)$ tuottaa U. Tämä yhdessä tiedon $\neg\forall x : kaava(x) \Leftrightarrow \exists x : \neg kaava(x)$ kanssa takaa, että $\neg\forall x : kaava(x)$ tuottaa aina saman totuusarvon kuin $\exists x : \neg kaava(x)$.

Yhtätotuuslain käyttösääntö (sivu 42) oikeuttaa käyttämään mitä tahansa $\equiv$-muotoista lakia isomman kaavan sisällä. Koska edellisessä laissa $kaava(x)$ edustaa mitä tahansa kaavaa, se voi edustaa mitä tahansa ¬:lla alkavaa kaavaa. Kirjoitamme sen $\neg kaava(x)$. Kun se sijoitetaan $kaava(x)$:n tilalle ja käytetään lakia $\neg\neg P \equiv P$, saadaan $\neg\forall x : \neg kaava(x) \equiv \exists x : \neg\neg kaava(x) \equiv \exists x : kaava(x)$. Siitä seuraa $\forall x : \neg kaava(x) \equiv \neg\neg\forall x : \neg kaava(x) \equiv \neg\exists x : kaava(x)$.

Olemme perustelleet seuraavat kaksi lakia sekä alla olevassa muodossa että siten, että ”$\Leftrightarrow$”:n tilalla on ”$\equiv$”:

$$\neg\forall x : kaava(x) \;\Leftrightarrow\; \exists x : \neg kaava(x)$$
$$\neg\exists x : kaava(x) \;\Leftrightarrow\; \forall x : \neg kaava(x)$$

Näitäkin kahta lakia kutsutaan De Morganin laeiksi. Ne muistuttavat sivulta 20 alkaen kerrottuja $\wedge$:n ja $\vee$:n De Morganin lakeja. Sivulla 22 kerrottu duaalisuus yleistyy kvanttoreihin siten, että jokainen $\forall$ ja $\exists$ vaihdetaan keskenään.

Se, että n ei ole jaollinen seitsemällä, ilmaistiin täällä kaavalla $\neg\exists x : n = 7x$. Symboleita ”div”, ”mod” ja ”$\forall$” ei saanut käyttää. Jos ”$\forall$” sallitaan, niin hieman lyhyempi kaava on mahdollinen. Miten?

Samalla tavalla kuin toinen De Morganin laki johdettiin toisesta, voidaan seuraavat lait johtaa luvussa 3.1 esitetyistä:

Huom! x ei saa esiintyä vapaana *kaava*:ssa!

$$\begin{aligned}
\forall x : \forall y : kaava(x,y) &\;\Leftrightarrow\; \forall y : \forall x : kaava(x,y)\\
\forall x : \big(kaava1(x) \wedge kaava2(x)\big) &\;\Leftrightarrow\; \big(\forall x : kaava1(x)\big) \wedge \big(\forall x : kaava2(x)\big)\\
\forall x : \big(kaava1(x) \vee kaava2(x)\big) &\;\Leftarrow\; \big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big)\\
\forall x : \big(kaava \vee kaava2(x)\big) &\;\Leftrightarrow\; kaava \vee \big(\forall x : kaava2(x)\big)\\
\forall x : kaava &\;\Leftrightarrow\; kaava
\end{aligned}$$

Huomaa, että keskimmäinen laki pätee vain oikealta vasemmalle. Osoita itse, että se ei päde kaksisuuntaisena, kirjoittamalla vastaesimerkki!

Pitää siis keksiä sellaiset kaavat $kaava1(x)$ ja $kaava2(x)$, eli MathCheckin kielelle käännettynä $P(x)$ ja $Q(x)$, että $\forall x : (P(x) \vee Q(x))$ on tosi, mutta $(\forall x : P(x)) \vee (\forall x : Q(x))$ ei ole. Jotta jälkimmäinen ehto toteutuisi, $P(x)$ ei saa olla aina tosi eikä $Q(x)$ saa olla aina tosi. Kuitenkin, jotta edellinen ehto toteutuisi, aina jommankumman pitää olla tosi. Siis $P(x)$:n pitää olla joskus epätosi, mutta silloin $Q(x)$:n pitää olla tosi. Samoin $Q(x)$:n pitää olla joskus epätosi, mutta silloin $P(x)$:n pitää olla tosi.

Sellaisia kaavapareja on vaikka kuinka paljon. Riittää valita $P(x)$:ksi jokin kaava, joka on toisinaan tosi ja toisinaan epätosi, ja $Q(x)$:ksi $\neg P(x)$. Voidaan siis valita vaikka $x = 0$ ja $x \neq 0$. Mutta $P(x)$ ja $Q(x)$ saavat olla yhtäaikaa tosi. Niinpä esimerkiksi $x > 2$ ja $x \leq 5$ kelpaavat.

Sivulla 28 kerrottiin, että sidotun muuttujan nimellä ei ole väliä, kunhan se ei sekaannu minkään muun muuttujan nimeen. Myöhemmin huomattiin, että sekaannusvaara saattaa syntyä esimerkiksi kun isomman kaavan osana oleva kaava korvataan yhtätodella kaavalla tai lauseke yhtäsuurella lausekkeella. Vaara voitiin kuitenkin välttää vaihtamalla jonkin sidotun muuttujan nimi. Nimen vaihtoa tarvitaan niin usein, että se ansaitsee tulla mainituksi omana lakinaan. Tietenkin uusi nimi täytyy valita siten, että se ei aiheuta sekaannusta. Tarkka ehto miten uuden nimen saa valita on hieman monimutkainen, mutta käytännön tarpeisiin riittää yleensä seuraava, yksinkertaisempi muotoilu:

Jos y ei esiinny lainkaan $kaava(x)$:ssä, niin
$\forall x : kaava(x) \Leftrightarrow \forall y : kaava(y)$ ja $\exists x : kaava(x) \Leftrightarrow \exists y : kaava(y)$.

Huomaatko lain muotoilussa jotakin kummallista? Jotakin, joka on omiaan herättämään epäilyn, että siinä on virhe?

Alemman rivin yhtäpitävyydet ovat symmetriset x:n ja y:n suhteen: niiden merkitykset eivät muutu, jos x ja y vaihdetaan keskenään. Vaihdos tekee ensimmäisestä yhtäpitävyydestä $\forall y : kaava(y) \Leftrightarrow \forall x : kaava(x)$. Mutta sivun 21 mukaan se tarkoittaa samaa kuin alkuperäinen yhtäpitävyys $\forall x : kaava(x) \Leftrightarrow \forall y : kaava(y)$. Samoin käy yhtäpitävyydelle $\exists x : kaava(x) \Leftrightarrow \exists y : kaava(y)$.

Mutta ensimmäisellä rivillä oleva ehto ei ole symmetrinen! Ei vaadita että x ei saa esiintyä $kaava(y)$:ssä, eikä se myöskään seuraa asetetuista vaatimuksista eli siitä, että y ei saa esiintyä $kaava(x)$:ssä. Jos esimerkiksi $kaava(x)$ on $x > 0 \wedge \forall x : x^2 \neq -1$, niin se toteuttaa vaatimuksen, mutta $kaava(y)$ on $y > 0 \wedge \forall x : x^2 \neq -1$, ja x esiintyy siinä. Jos huomasit tämän epäsymmetrian, niin onnittele itseäsi!

Tutkikaamme asiaa tarkemmin. Lain ensimmäinen rivi voidaan lieventää sallimaan y:n esiintymiä $kaava(x)$:ssä, kunhan esiintymät ovat sidottuja eikä x esiinny vapaana missään kohdassa, jossa jokin sidottu y on olemassa. Jälkimmäisen ehdon vuoksi jokainen x:n vapaa esiintymä $kaava(x)$:ssä muuttuu y:n vapaaksi (eikä sidotuksi) esiintymäksi $kaava(y)$:ssä. Ensimmäisen ehdon vuoksi jokainen y:n vapaa esiintymä $kaava(y)$:ssä on peräisin jostakin x:n vapaasta esiintymästä $kaava(x)$:ssä (eikä siis ollut olemassa jo $kaava(x)$:ssä). Siksi x:n vapaat esiintymät $kaava(x)$:ssä ja y:n vapaat esiintymät $kaava(y)$:ssä vastaavat täsmälleen toisiaan. Kun ne sidotaan kvanttoreilla, saadaan yhtäpitävät kaavat.

Vaikka tämä lievennetty ehto ei näytä symmetriseltä, se on symmetrinen. Nimittäin jos se pätee, niin myös symmetrinen ehto pätee, kuten kohta näemme. Koska $kaava(y)$ muodostetaan korvaamalla jokainen x:n vapaa esiintymä y:llä, ei x esiinny vapaana $kaava(y)$:ssä. Niissä $kaava(y)$:n kohdissa, joissa jokin sidottu x on olemassa, ei y esiinny vapaana seuraavista syistä. Niissä y ei esiintynyt vapaana $kaava(x)$:ssä, koska y ei esiintynyt lainkaan vapaana $kaava(x)$:ssä. Niihin ei syntynyt y:n esiintymiä sijoituksessa, koska niissä kohdissa x oli sidottu, joten sen esiintymät eivät korvaantuneet y:llä.

Siis epäilyttävä epäsymmetria katoaa, kun muuttujan uutta nimeä koskevat vaatimukset analysoidaan tarkemmin. Mutta niin muotoiltu laki on monimutkainen. Yksinkertaisempi laki hukkaa mahdollisuuksia olemalla toisinaan antamatta lupaa sellaisen nimen käyttöön, jota todellisuudessa voisi käyttää. Mutta se ei ole olennainen ongelma, koska eri nimet eivät hevin lopu kesken.

Sen, että yksinkertaisempi laki on pätevä, saa siitä, että monimutkaisempi laki on pätevä. Nimittäin jos yksinkertaisemman lain ehto toteutuu eli y ei esiinny lainkaan $kaava(x)$:ssä, niin myös monimutkaisemman lain ehto toteutuu eli y:n kaikki esiintymät $kaava(x)$:ssä ovat sidottuja eikä x esiinny vapaana missään kohdassa,

jossa jokin sidottu y on olemassa.

Seuraavaksi tarkastelemme tapausta, jossa erilaiset kvanttorit ovat peräkkäin. Olkoon puheenaiheena ihmiset. Tarkoittakoon $\ddot{A}(x,y)$, että x on tai oli y:n äiti. Koska jokaisella ihmisellä on tai on ollut äiti, on $\forall y : \exists x : \ddot{A}(x,y)$ tosi. Mutta kukaan ei ole eikä ole ollut kaikkien ihmisten äiti. Siksi $\exists x : \forall y : \ddot{A}(x,y)$ ei ole tosi.

Sen sijaan vastakkaiseen suuntaan pätee riippumatta puheenaiheesta ja siitä, mitä $\ddot{A}(x,y)$ tarkoittaa. Esimerkiksi jos työpaikalla joku on ystävällinen kaikille (myös itselleen), niin siinä työpaikassa jokaisella on joku, joka on hänelle ystävällinen (jollei kukaan muu, niin ainakin tämä kaikille ystävällinen henkilö).

Yleisemmin pohdittuna, $\exists x : \forall y : \mathit{kaava}(x,y)$ tarkoittaa, että on olemassa sellainen x:n arvo, että millä tahansa y:n arvolla pätee $\mathit{kaava}(x,y)$. Siinä x valitaan ennen kuin y, joten x saa yhden arvon, ja tämä yksi arvo kelpaa jokaiselle y. Toisaalta $\forall y : \exists x : \mathit{kaava}(x,y)$ sanoo, että y valitaan ennen kuin x, ja x saadaan valita erikseen kullekin y:n arvolle. Siinä tapauksessa x voi saada eri arvoja y:stä riippuen. Kuitenkin, jos kaikille y kelpaava yhteinen x:n arvo on olemassa, niin sitä saa käyttää, sillä lupa valita x:lle eri arvoja on lupa eikä määräys. Jos $\exists x : \forall y : \mathit{kaava}(x,y)$ pätee, niin sellainen y:n arvo on olemassa. Siksi seuraava on logiikan laki:

$$\exists x : \forall y : \mathit{kaava}(x,y) \;\Rightarrow\; \forall y : \exists x : \mathit{kaava}(x,y)$$

Laajalti tiedetään, että esimerkiksi $2x^2+7x+0 = 2x^2+7x$ on oikein siksi, että mikään luku ei muutu kun siihen lisätään nolla. Vähemmän kuitenkin puhutaan siitä, mitä tapahtuu, kun edetään siitä, että mikään luku ei muutu kun siihen lisätään nolla, tulokseen $2x^2+7x+0 = 2x^2+7x$. Nimittäin jotakin siinä täytyy tapahtua, koska "$2x^2+7x$" ei ole luku vaan lauseke. Se, että mikään luku ei muutu kun siihen lisätään nolla, ilmaistaan usein kaavalla $\forall a : a+0 = a$. Seuraava laki sallii johtaa siitä $2x^2+7x+0 = 2x^2+7x$. Laki ilmaisee sen, että jos $\mathit{kaava}(x)$ on tosi millä tahansa x:n arvolla, niin se on tosi myös sillä arvolla, jonka *lauseke* tuottaa. Annamme sille jatkoa varten nimen K1.

> K1: Jos jokaisessa mahdollisessa tilanteessa *lauseke* on määritelty, ja jos mikään sen vapaa muuttuja ei joudu sidotuksi $\mathit{kaava}(\mathit{lauseke})$:ssa, niin $\forall x : \mathit{kaava}(x) \;\Rightarrow\; \mathit{kaava}(\mathit{lauseke})$.

Lain alussa oleva ehto varmistaa, että *lauseke* todella tuottaa jonkin arvon. Ilman sitä voitaisiin muun muassa todesta kaavasta $\forall x : x = x$ johtaa määrittelemätön kaava $\frac{1}{0} = \frac{1}{0}$. Jälkimmäinen lain alussa olevista ehdoista estää samannimisiä muuttujia sekoittumasta toisiinsa. Se estää esimerkiksi johtamasta todesta kaavasta $\forall x : \exists y : y > x$ epätoden kaavan $\exists y : y > y$.

Tämä laki on siis käytössä aina kun puheenaiheen arvoille luvattuja yleisiä ominaisuuksia sovelletaan. Sitä käytetään kun siitä, että aina ohjelman rivin 2

alussa $1 \leq a \leq y \leq n+1$, päätellään, että viimeisellä kerralla rivin 2 alussa $1 \leq a \leq y \leq n+1$. Sitä käytetään kun siitä, että jokainen auto menee joskus rikki, päätellään että Mikan faijan auto menee joskus rikki. Sitä käytetään siis hyvin paljon — tosin usein tiedostamatta.

Vastaavaa $\exists$:n lakia käytetään paljon vähemmän. Mutta ei sekään täysin tarpeeton ole, vaan sitäkin käytetään päättelyissä toisinaan.

> O1: Jos jokaisessa mahdollisessa tilanteessa *lauseke* on määritelty, ja jos mikään sen vapaa muuttuja ei joudu sidotuksi *kaava*(*lauseke*):ssa, niin $kaava(lauseke) \Rightarrow \exists x : kaava(x)$.

Matematiikassa, tietojenkäsittelytieteessä ja niin edelleen on tavallista johtaa lakeja siten, että valitaan jotkin muuttujat edustamaan mitä tahansa puheenaiheen arvoja. Päättelyn aikana niitä käsitellään vapaina muuttujina, ja lopputuloksen ajatellaan pätevän kaikille puheenaiheen arvoille. Esimerkiksi valitaan a ja b edustamaan mielivaltaisia reaalilukuja, lasketaan $(a+b)(a-b) = a^2 - ab + ba - b^2 = a^2 - ab + ab - b^2 = a^2 - b^2$, ja todetaan että $(a+b)(a-b) = a^2 - b^2$.

Logiikassa on tavallista toimia muuten samalla tavalla, mutta lopputulokseen lisätään kaikkikvanttorit. Siten esimerkistämme saadaan $\forall a : \forall b : (a+b)(a-b) = a^2 - b^2$. Kaikkikvanttorit kertovat, että tulos pätee kaikille reaaliluvuille eikä vain niille, jotka sillä hetkellä sijaitsevat muuttujissa a ja b. Niiden ansiosta tulosta voi käyttää myöhemmissä päättelyissä soveltamalla edellä esitettyä lakia K1. Voidaan esimerkiksi sijoittaa a:n ja b:n tilalle $2x$ ja 5, ja päätellä, että $(2x+5)(2x-5) = (2x)^2 - 5^2$.

Kaikkikvanttorin lisääminen perustuu seuraavaan lakiin:

> K2: Jos $kaava(x)$ on johdettu päättelyllä, jonka lähtökohdissa x ei esiinny vapaana, niin $\forall x : kaava(x)$.

K1 ja K2 ovat kaksi tärkeintä kaikkikvanttorin lakia. Muut kaikkikvanttorin lait, tai ainakin jokainen niiden käyttötapaus, voidaan johtaa niistä. Esimerkiksi $\big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big) \Rightarrow \forall x : \big(kaava1(x) \vee kaava2(x)\big)$ voidaan johtaa seuraavasti. Apuna tarvitaan seuraavat "$\vee$":n lait:

T1: $kaava1 \Rightarrow kaava1 \vee kaava2$

T2: $kaava2 \Rightarrow kaava1 \vee kaava2$

T3: Jos $kaava1 \Rightarrow kaava3$ ja $kaava2 \Rightarrow kaava3$, niin $kaava1 \vee kaava2 \Rightarrow kaava3$.

Ensin johdetaan $\forall x : kaava1(x) \Rightarrow \forall x : \big(kaava1(x) \vee kaava2(x)\big)$:

$$
\begin{array}{lll}
 & \forall x : kaava1(x) & \text{lähtökohta, jossa } x \text{ ei vapaa} \\
\Rightarrow & kaava1(x) & \mathsf{K1} \\
\Rightarrow & kaava1(x) \vee kaava2(x) & \mathsf{T1} \\
\Rightarrow & \forall x : \big(kaava1(x) \vee kaava2(x)\big) & \mathsf{K2}
\end{array}
$$

Sitten samalla tavalla johdetaan $\forall x : kaava2(x) \Rightarrow \forall x : \big(kaava1(x) \vee kaava2(x)\big)$. Siihen tarvitaan T2 (eikä T1). Niistä T3 tuottaa luvatun lain $\big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big) \Rightarrow \forall x : \big(kaava1(x) \vee kaava2(x)\big)$.

Jälleen kerran on hyödyllistä tarkastella, mikä menee pieleen, jos samaan tapaan yritetään johtaa jotakin, joka ei pidä paikkaansa. Mikä seuraavassa on vialla? Ensin K1:llä johdetaan $\forall x : \big(kaava1(x) \vee kaava2(x)\big) \Rightarrow kaava1(x) \vee kaava2(x)$. Sitten käsitellään kumpikin tapaus erikseen K2:n ja T1:n tai T2:n avulla:

$$
\begin{array}{l}
kaava1(x) \;\Rightarrow\; \forall x : kaava1(x) \;\Rightarrow\; \big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big) \\
kaava2(x) \;\Rightarrow\; \forall x : kaava2(x) \;\Rightarrow\; \big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big)
\end{array}
$$

Niistä T3 tuottaa $kaava1(x) \vee kaava2(x) \Rightarrow \big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big)$. Kaikkiaan saimme $\forall x : \big(kaava1(x) \vee kaava2(x)\big) \Rightarrow \big(\forall x : kaava1(x)\big) \vee \big(\forall x : kaava2(x)\big)$. Mutta ei olisi pitänyt saada, sillä, kuten näimme aikaisemmin, se ei ole pätevä.

Vika oli K2:n soveltamisissa. Niissä lähtökohtina olivat $kaava1(x)$ ja $kaava2(x)$. Niissä x esiintyy vapaana, joten K2:n jos-osaa rikottiin.

K1 ja O1 ovat selvästi toistensa vastinparit. Näin ollen voisi odottaa, että myös K2:lla on vastinpari. Sillä on, mutta se ei ole yhtä yksinkertainen kuin K2.

> O2: Jos $kaava1(y) \Rightarrow kaava$ on johdettu päättelyllä, jonka lähtökohdissa y ei esiinny, ja jos y ei esiinny myöskään $kaava$:ssa eikä $(\exists x : kaava1(x))$:ssä, niin $\exists x : kaava1(x) \Rightarrow kaava$.

Käytännössä tämä laki tarkoittaa sitä, että sille arvolle, jonka $\exists x : kaava1(x)$ väittää olevan olemassa, saa antaa tilapäisen nimen y ja käyttää sitä päättelyssä. Koska y:n arvo tulee $\exists$:sta, siitä ei voida luvata mitään muuta kuin että $kaava1(y)$ pätee, joten y ei saa esiintyä vapaana voimassa olevissa oletuksissa. Koska y on tilapäinen, se ei saa esiintyä vapaana johtopäätöksessä eli $kaava$:ssa. Se ei saa myöskään sekoittua kaavan $\exists x : kaava1(x)$ muuttujiin. Tämän sanomiseen ei riitä vapaiden esiintymien kielto, mutta kaikkien esiintymien kielto riittää. Kaikkien esiintymien kielto on myös selkeä eikä juurikaan haittaa O2:n käyttöä, sillä nimiä riittää.

O2:n mukainen päättelytapa on tavallinen matematiikassa ja tietojenkäsittelytieteessä. Se toteutetaan kuitenkin useimmiten sanallisesti eikä kirjoittamalla ”$\exists$” näkyviin ja vetoamalla O2:een.

O1 ja O2 ovat kaksi tärkeintä olemassaolokvanttorin lakia. Muut olemassaolokvanttorin lait, tai ainakin jokainen niiden käyttötapaus, voidaan johtaa niistä.

Merkinnät $\forall x\,;\,rajoite(x) : kaava(x)$ ja $\exists x\,;\,rajoite(x) : kaava(x)$ eivät tarvitse pitkää käsittelyä. Vaikka sidottu muuttuja saa niissä vain osan puheenaiheen arvoista, ne voidaan määritellä tavallisten kvanttorimerkintöjen avulla seuraavasti:

$$\begin{aligned} \forall x\,;\,rajoite(x) : kaava(x) &\quad\Leftrightarrow\quad \forall x : \neg\, rajoite(x) \vee kaava(x) \\ \exists x\,;\,rajoite(x) : kaava(x) &\quad\Leftrightarrow\quad \exists x : rajoite(x) \wedge kaava(x) \end{aligned}$$

Kaikkikvanttorista on hyvä tietää vielä yksi asia: jos kvantifioidulla muuttujalla voi olla vain äärellinen määrä eri arvoja, niin ”$\forall$” voidaan korvata ”$\wedge$”:lla. Tämä säilyy voimassa, jos muuttujalla voi olla äärettömästi eri arvoja, mutta niistä vain äärellisellä määrällä voi *kaava* tuottaa muun totuusarvon kuin T. Esimerkiksi kokonaisluvuista puhuttaessa $\forall i\,;\,2 \leq i \leq 5 : kaava(i)$ tarkoittaa samaa kuin $kaava(2) \wedge kaava(3) \wedge kaava(4) \wedge kaava(5)$.

3.4 Prinsessatehtävä

Ylen uutisessa 25.9.2024 kerrottiin, että uusi tekoälyjärjestelmä oli ratkaissut seuraavan tehtävän 30 sekunnissa:

> Prinsessa on yhtä vanha kuin prinssi tulee olemaan, kun prinsessa on kaksi kertaa niin vanha kuin prinssi oli silloin, kun prinsessan ikä oli puolet heidän nykyisen iän summasta. Mikä on prinssin ja prinsessan ikä?

Uutisessa sanottiin ”Ihmiselle voi olla vaikeaa hahmottaa pelkkää kysymystä, vaikka tiedossa olisi oikea vastaus.” Tehtävä on kuitenkin ihmistenkin melko helposti hahmotettavissa kääntämällä se kaavoiksi. Teemme niin. Sen jälkeen ratkaisemme sen MathCheckin avustamana vaihe vaiheelta edeten.

Mainittakoon, että ratkaisemme sen sillä tavalla, koska tavoitteenamme on harjoitella logiikkaa, ja koska tämä kurssi yrittää olla edellyttämättä lukion pitkää matematiikkaa esitiedoikseen. Jos tavoitteemme olisi ratkaista se mahdollisimman kätevästi ja jos olisi lupa käyttää lukion pitkän matematiikan tietoja, niin sen voisi ratkaista myös täysin mekaanisesti vähällä vaivalla.

Tehtävä ratkaistaan täällä.

3.5 Hieman toisen kertaluvun logiikasta

$$\neg\exists P : P(u) \wedge \neg P(v) \wedge \forall x : \forall y : \neg P(x) \vee \neg(x \rightsquigarrow y) \vee P(y)$$

4 Kaavojen toteuttaminen tietokoneessa DFA:illa

miksi $1 = 2$ ja $0x = 1$ ovat olennaisesti sama kuin F

Kokonaislujen moodissa MathCheck valitsee sellaisen palautteen, jonka esitys bittijonona (lopun päättymätön 0-jono pois lukien) on mahdollisimman lyhyt. Lyhin on tyhjä bittijono (siis loputtomasti pelkkää nollaa). Siinä jokainen muuttuja saa arvon 0. Seuraavaksi lyhimmässä ensimmäinen bitti on 1 ja muut bitit nollia. Siinä ensimmäinen muuttuja saa arvon -1 ja muut arvon 0.

Jos MathCheckin antama palaute on niin iso, että se ei mahdu palautelaatikkoon, niin kannattaa klikata ”uuteen”-nappia. Silloin palaute ilmestyy uuteen välilehteen tai uuteen selainikkunaan riippuen selaimen asetuksista. Nappeja ”oikealle” ja ”uuteen” saa klikata samalla syötteellä niin monta kertaa kuin haluaa ja missä järjestyksessä tahansa.

5 Päätteleminen

6 Kaavojen totuutta ei voi pelkistää kaavaksi

Quinen paradoksi (1962):

> ”Laitettuna lainausmerkeissä itsensä eteen tuottaa epätoden” laitettuna lainausmerkeissä itsensä eteen tuottaa epätoden.

Lämmittelyhavainto: luonnollisten lukujen kaikkia aina määriteltyjä yhden vapaan muuttujan kaavoja $P_1(n)$, $P_2(n)$, … ei voi esittää yhtenä kahden vapaan muuttujan kaavana $Q(i,n)$ siten, että $Q(i,n)$ tuottaa aina saman totuusarvon kuin $P_i(n)$ eli $Q(i,n) \Leftrightarrow P_i(n)$. Nimittäin jos sellainen kaava olisi olemassa, niin olkoon $R(n)$ se kaava, joka saadaan $Q(i,n)$:stä käyttämällä myös i:n paikalla n:ää ja lisäämällä ympärille sulkeet ja eteen ”$\neg$”. Koska $R(n)$ on yhden vapaan muuttujan kaava, on olemassa sellainen luonnollinen luku r, että $R(n) \Leftrightarrow Q(r,n)$. Toisaalta rakentamistapansa vuoksi $R(n) \Leftrightarrow \neg Q(n,n)$. Kun n:n arvoksi valitaan r saadaan ensimmäisestä $R(r) \Leftrightarrow Q(r,r)$ ja toisesta $R(r) \Leftrightarrow \neg Q(r,r)$, joten $Q(r,r) \Leftrightarrow \neg Q(r,r)$, joka on ristiriita.

7 Aksiomatisointi, tulkinta ja malli

Kahvin, teen, maidon ja sokerin aksioomat:

- $\neg(K \wedge T)$
- $\neg(M \vee S) \vee K \vee T$

Sivulla 58 ilmaistiin saavutettavuus seuraavasti toisen kertaluvun kaavana:

$$\neg\exists P : P(u) \wedge \neg P(v) \wedge \forall x : \forall y : \neg P(x) \vee \neg(x \rightsquigarrow y) \vee P(y)$$

Oletamme, että käytössä on ensimmäisen kertaluvun tai sitä laajempi logiikka; kukin aksiooma on kaava; numeroituvasti äärettömät aksioomajoukot on sallittu; kukin formaali päättely koostuu äärellisestä määrästä askelia; ja kukin päättelyaskel voi käyttää vain äärellistä määrää aksioomia. Tavoitteena on osoittaa metatulos, että joko käsitettä "saavutettavuus" ei voi määritellä, aksioomien kaikkia loogisia seurauksia ei voi päätellä, tai päättelyjärjestelmä ei ole totuudensäilyttävä.

Käytämme kieltä, jossa on ei-looginen symboli "$\rightsquigarrow$". Se edustaa 2-paikkaista relaatiota. Epämuodollisesti voi ajatella, että puheenaiheena on jokin epätyhjä suunnattu graafi, ja $u \rightsquigarrow v$ tarkoittaa, että u:sta on kaari v:hen.

Tarkoittakoon $u \rightsquigarrow^* v$ kaavaa, joka sanoo, että v on saavutettavissa u:sta. Arkimatematiikan kielellä "$\rightsquigarrow^*$" on "$\rightsquigarrow$":n refleksiivinen transitiivinen sulkeuma. Jos sellainen kaava voidaan kirjoittaa, niin voimme asettaa seuraavat kaksi aksioomaa. Jälkimmäinen sanoo, että graafi ei haaraudu eteenpäin.

$$\begin{aligned} \forall u : \forall v : \quad & u \rightsquigarrow^* v \\ \neg\exists u : \exists v_1 : \exists v_2 : \quad & u \rightsquigarrow v_1 \wedge u \rightsquigarrow v_2 \wedge \neg(v_1 = v_2) \end{aligned}$$

Tämän aksioomajoukon mallit ovat yksittäinen solmu josta ei ole kaarta itseensä, sekä yhden pituinen silmukka, kahden pituinen silmukka, kolmen pituinen silmukka ja niin edelleen. Toisin sanoen, mallit ovat graafit muotoa $v_0 \rightsquigarrow v_1 \rightsquigarrow \ldots \rightsquigarrow v_n$, missä $n \in \mathbb{N}$, $v_0 = v_n$, ja $v_i \neq v_j$ kun $0 \leq i < j < n$. Muita malleja ei ole, sillä jos solmusta ei ole yhtään kaarta eteenpäin, niin siitä ei pääse muihin solmuihin, ja jos jokaisesta solmusta on täsmälleen yksi kaari eteenpäin mutta $v_0 \rightsquigarrow v_1 \rightsquigarrow v_2 \rightsquigarrow \ldots$ ei lopulta palaa v_0:aan tai ei sisällä solmua u, niin $v_1 \not\rightsquigarrow^* v_0$ tai $v_1 \not\rightsquigarrow^* u$.

Otamme käyttöön lyhenteen $u \hookrightarrow v$ tarkoittamaan $u = v \vee u \rightsquigarrow v$. Esimerkiksi $\exists w_1 : \exists w_2 : u \hookrightarrow w_1 \hookrightarrow w_2 \hookrightarrow v$ sanoo, että v on saavutettavissa u:sta enintään kolmella kaarella. Lisäämme alla näytetyn äärettömän aksioomajoukon. Aksiooma A_i sanoo, että on olemassa sellaiset kaksi solmua, että jälkimmäinen ei ole saavutettavissa ensimmäisestä enintään $i+1$ kaarella. Se sulkee pois ne mallit $v_0 \rightsquigarrow \ldots \rightsquigarrow v_n$, joissa $n \leq i+2$.

$$\begin{array}{lllll} \mathsf{A}_1 & \exists u : \exists v : & \neg\exists w_1 : & & u \hookrightarrow w_1 \hookrightarrow v \\ \mathsf{A}_2 & \exists u : \exists v : & \neg\exists w_1 : \exists w_2 : & & u \hookrightarrow w_1 \hookrightarrow w_2 \hookrightarrow v \\ \mathsf{A}_3 & \exists u : \exists v : & \neg\exists w_1 : \exists w_2 : \exists w_3 : & & u \hookrightarrow w_1 \hookrightarrow w_2 \hookrightarrow w_3 \hookrightarrow v \\ & \vdots & & \vdots & \vdots \end{array}$$

Laajennetulla aksioomajoukolla ei siis ole malleja. Jos kaikki loogiset seuraukset voidaan päätellä, niin laajennetusta aksioomajoukosta voidaan päätellä ristiriita.

Koska päättelyt ovat äärellisiä ja kukin päättelyaskel käyttää vain äärellistä määrää aksioomia, on olemassa sellainen n, että ristiriidan päättely ei käytä mitään aksioomista A_n, A_{n+1}, A_{n+2}, ... Siksi päättely menee läpi vaikka ne jätettäisiin pois. Ristiriita voidaan siis päätellä myös aksioomajoukolle, jolle $v_0 \rightsquigarrow \ldots \rightsquigarrow v_{n+2}$ on malli. Niinpä päättelyjärjestelmä ei ole totuudensäilyttävä.

aksiooma, aksioomaskeema, looginen seuraus

8 Gödelin täydellisyyslause

9 Tietokoneella tarkastettavuuden tärkeys

10 Gödelin epätäydellisyyslauseet

11 NP-täydellisyys

12 Lopuksi

13 Symbolien ja termien selityksiä ja suomennoksia

$\neg$ Katso negaatio 76.

$\wedge$ Katso konjunktio 69.

$\vee$ Katso disjunktio 65.

$\rightarrow$ Katso materiaalinen implikaatio 74.

$\leftrightarrow$ Katso materiaalinen ekvivalenssi 74.

$\forall$ Katso kvanttori 70.

$\exists$ Katso kvanttori 70.

$\Rightarrow$ Katso päättelyimplikaatio 80.

$\Leftrightarrow$ Katso yhtäpitävyys 92.

$\equiv$ Katso yhtätotuus 93.

$\models$ Katso tulkinta 90 ja looginen seuraus 71.

$\vdash$ Katso päättely 79.

φ Kreikkalainen kirjain ”fii”. Usein edustaa mitä tahansa kaavaa.

ψ Kreikkalainen kirjain ”psii”. Usein edustaa mitä tahansa kaavaa.

χ Kreikkalainen kirjain ”khii”. Usein edustaa mitä tahansa kaavaa.

abbreviation $\rightarrow$ lyhenne 73

aksiomatisointi Katso aksiooma 62.

aksiooma Puheenaiheen *aksiomatisointi* on joukko suljettuja kaavoja, jotka yhdessä ilmaisevat puheenaiheen ominaisuudet jossakin laajuudessa. Sen yksittäiset kaavat ovat *aksioomia*. Esimerkkejä reaalilukujen aksioomista ovat $\forall a : a + 0 = a$ ja $\forall a : \forall b : a + b = b + a$. Aksiomatisoinnissa voi olla äärettömästi kaavoja, katso aksioomaskeema 63.

Sitä mitä tässä tekstissä kutsutaan aksiomatisoinniksi, kutsutaan logiikan kirjallisuudessa usein *teoriaksi*. Teoriaksi kutsutaan usein myös aksioomia ja niistä pääteltävissä olevia suljettuja kaavoja yhdessä. ”Teoria” näissä merkityksissä on eri asia kuin se mitä arkikielessä tarkoitetaan sanalla ”teoria”.

Siitä aiheutuu sekaannuksen vaara. Siksi tässä tekstissä sanaa ”teoria” käytetään arkikielen merkityksessä, ja ominaisuuksien ilmaisemiseen käytettävää suljettujen kaavojen joukkoa kutsutaan aksiomatisoinniksi.

Ensimmäisen kertaluvun tapauksessa aksiomatisointi on *täydellinen*, jos ja vain jos se kattaa kaikki ominaisuudet, jotka ovat ilmaistavissa suljettuina kaavoina käytössä olevilla ei-loogisilla symboleilla (katso ei-looginen symboli 65). Tarkemmin sanoen, jokainen valitun nimikirjoituksen mukainen suljettu ensimmäisen kertaluvun kaava voidaan päätellä joko itse, tai sen negaatio voidaan päätellä. Käytössä on paljon epätäydellisiä aksiomatisointeja joko siksi, että tarkoitus on esittää useiden samankaltaisten järjestelmien (esimerkiksi kaikkien kuntien) yhteiset ominaisuudet tai siksi, että täydellinen aksiomatisointi on mahdotonta (katso tosi aritmetiikka 88).

Aksiomatisointi on *ristiriitainen*, jos ja vain jos siitä voidaan päätellä sekä jokin kaava että sen negaatio. ”Täydellinen” sellaisena kuin se määriteltiin edellä ei sisällä ristiriidattomuuden vaatimusta. Ristiriitainen aksiomatisointi on valtavirran logiikoissa hyödytön, koska siitä voidaan päätellä kaikki kaavat (katso mahdottomasta seuraa mitä tahansa 73). Siksi sanaa ”täydellinen” käytetään usein myös merkityksessä ”täydellinen ja ristiriidaton”.

Toisen kertaluvun tapauksessa aksiomatisoinnin täydellisyyttä ei ole mielekästä määritellä kuten edellä, koska toisen kertaluvun logiikassa päättely on epätäydellistä. Tämä ongelma olisi kierrettävissä käyttämällä määritelmässä päättelyn sijaan loogista seurausta. Niin ei kuitenkaan yleensä tehdä. Siihen ei ole suurta tarvetta, koska toisen kertaluvun aksiomatisoinnilla voidaan toisinaan saavuttaa vielä vahvempi ominaisuus, katso kategorinen 68.

aksioomaskeema Ilmaus, joka esittää useita aksioomia. Tyypillisesti aksioomaskeema esittää äärettömän monta aksioomaa äärellisellä ilmauksella. Esimerkiksi luonnollisiin lukuihin voidaan lisätä uusi luku ottamalla käyttöön vakiosymboli c ja aksioomaskeema $c \neq 0$, $c \neq 1$, $c \neq 2$, ... (katso tosi aritmetiikka 88).

Tavallisesti vaaditaan, että aksioomaskeema on rekursiivinen, eli että ainakin periaatteessa on olemassa tietokoneohjelma, joka tarkastaa mistä tahansa äärellisestä merkkijonosta äärellisessä ajassa, onko se skeemaan kuuluva aksiooma.

alkio Sanalla ”alkio” voidaan matematiikassa ja logiikassa viitata melkeinpä mihin tahansa. Alkiot voivat olla lukuja, ihmisiä, keittiövälineitä ja niin edelleen. Olennaista on, että eri alkioita voidaan koota kokoelmiksi, joita kutsutaan *joukoiksi*. Tämän tekstin tarpeisiin riittää ajatella, että joukko tarkoittaa

sitä mitä se tarkoittaa arkikielessäkin, sillä erotuksella, että joukossa voi olla myös hyvin vähän alkioita, jopa ei yhtään (*tyhjä joukko*). Joukosta saadaan *osajoukko* poistamalla siitä nolla tai useampia alkioita (vaikka kaikki).

arity → paikkaluku, katso nimikirjoitus 76.

assignment → tilanne 88

associative → liitännäinen 71

atomic formula → atomikaava 64

atomikaava Propositiologiikan atomikaavoja ovat F, T ja kukin propositiomuuttuja.

Ensimmäisen kertaluvun logiikan atomikaavoja ovat F, T, kaikki ilmaukset muotoa *termi*1 = *termi*2, sekä kaikki ilmaukset, jotka on muodostettu asettamalla relaatiosymbolin syötteeksi paikkaluvun mukainen määrä termejä. Katso nimikirjoitus 76 ja termi 87.

Toisen kertaluvun logiikan atomikaavoja ovat lisäksi kaikki ilmaukset, jotka on muodostettu asettamalla relaatiomuuttujan tai kyllä/ei-ominaisuuksia edustavan muuttujan syötteeksi paikkaluvun mukainen määrä termejä. Termin käsite laajenee kattamaan myös funktioita edustavien muuttujien käytön.

avoin kaava Kaava, jossa esiintyy ainakin yksi vapaa muuttuja. Avoimen kaavan totuusarvo riippuu tyypillisesti vapaille muuttujille annettavista arvoista (katso tilanne 88). Välttämätöntä tämä ei kuitenkaan ole, sillä esimerkiksi $x = 3 \wedge 1 = 2$ on aina epätosi riippumatta siitä, minkä arvon x saa. Avoimen kaavan totuusarvo riippuu tyypillisesti myös tulkinnasta eli siitä, mitä siinä esiintyvät ei-loogiset symbolit (esimerkiksi ”3”, ”+” ja ”<”) tarkoittavat (katso tulkinta 90).

axiom → aksiooma 62

axiom schema → aksioomaskeema 63

axiomatization → aksiomatisointi, katso aksiooma 62.

biconditional Katso materiaalinen ekvivalenssi 74.

bound variable → sidottu muuttuja 85

categorical → kategorinen 68

Church–Turing teesi Katso rekursiivinen 83.

Church–Turing thesis → Church–Turing teesi, katso rekursiivinen 83.

closed formula → suljettu kaava 87

commutative → vaihdannainen 92

compactness theorem → kompaktisuuslause 69

complete axiomatization / theory → täydellinen aksiomatisointi / teoria, katso aksiooma 62.

conjunction → konjunktio 69

consistent → ristiriidaton 85

constant symbol → vakiosymboli, katso nimikirjoitus 76.

contradiction → ristiriita 85

countable set → numeroituva joukko 77

counterexample → vastaesimerkki 92

deduction theorem Jos joukosta kaavoja täydennettynä kaavalla *kaava*1 saa päätellä *kaava*2, niin alkuperäisestä joukosta saa päätellä *kaava*1 → *kaava*2. Katso myös modus ponens 75.

disjunction → disjunktio 65

disjunktio Symboli ”$\vee$” tai kaava muotoa *kaava*1 $\vee$ *kaava*2. Jälkimmäinen on tosi täsmälleen silloin kun joko *kaava*1 tai *kaava*2 tai molemmat on tosi.

domain of discourse → puheenaihe 78

efektiivinen Katso rekursiivinen 83.

effective → efektiivinen, katso rekursiivinen 83.

ei-looginen symboli Symboli, joka ottaa syötteekseen puheenaiheen alkion tai alkioita (paitsi ehkä ”=”) taikka vakiosymboli, jonka arvo on puheenaiheen alkio. Katso nimikirjoitus 76. Vastakohtana ovat *loogiset symbolit*, kuten $\wedge$, $\forall$ ja muuttujat. Symboli ”=” katsotaan usein loogiseksi symboliksi, mutta se voidaan katsoa myös ei-loogiseksi.

ekvivalenssi Kahta kohdetta toisiinsa vertaava toiminto, joka edustaa jonkinlaista samanlaisuutta tai samankaltaisuutta. Tunnetuin ekvivalenssi on koulumatematiikan ”=”. Ihmisten välisestä ekvivalenssista esimerkiksi kelpaa ”ovat samassa urheilun ikäluokkasarjassa”.

Ekvivalenssit noudattavat kolmea lakia: Jokainen olio on samankaltainen itsensä kanssa. Jos x on samankaltainen y:n kanssa, niin y on samankaltainen x:n kanssa. Jos x on samankaltainen y:n kanssa ja y on samankaltainen z:n kanssa, niin x on samankaltainen z:n kanssa.

Ekvivalenssi ei välttämättä takaa, että jos samankaltaisille olioille tehdään sama toiminto, niin saadaan samankaltaiset lopputulokset. Esimerkiksi 47- ja 49-vuotias ihminen ovat samassa urheilun ikäluokkasarjassa, nimittäin sarjassa 45–49-vuotiaat; mutta kun kumpikin on vanhentunut yhden vuoden, he ovat eri ikäluokkasarjassa.

Logiikassa on käytössä muitakin ekvivalensseja kuin ”=”. Katso materiaalinen ekvivalenssi 74 ja yhtäpitävyys 92.

ensimmäinen kertaluku Ilmauksella ”ensimmäinen kertaluku” tarkoitetaan tavallisesti ensimmäisen kertaluvun logiikkaa, eli predikaattilogiikkaa, jossa muuttujat voivat saada arvoikseen vain puheenaiheen joukon alkioita. Ensimmäisen kertaluvun logiikka on ylivoimaisesti tärkein predikaattilogiikka, koska siinä totuus (tarkemmin sanoen, looginen seuraus) ja todistettavuus ovat sama asia (katso looginen seuraus 71 ja Gödelin täydellisyyslause 67).

Toisaalta ensimmäisen kertaluvun ilmaisuvoima on jossain määrin rajallinen: on monia tärkeitä väitteitä, joita ei voi esittää ensimmäisen kertaluvun kaavoina. Ilmaisuvoiman ja todistettavuuden välillä on pakko valita jokin kompromissi (katso toinen kertaluku 88, Gödelin epätäydellisyyslauseet 67 ja kompaktisuuslause 69).

Korkeamman kertaluvun logiikoissa on myös muunlaisia sidottuja muuttujia. Korkeamman kertaluvun logiikoista tärkein on toisen kertaluvun logiikka (katso toinen kertaluku 88).

Entscheidungsproblem Katso ratkeamattomuus 81.

equivalence → ekvivalenssi 66

excluded middle, law of → kolmannen poissuljetun laki 69

field → kunta 69

first order → ensimmäinen kertaluku 66

formula → kaava 68, katso myös atomikaava 64 ja kvanttori 70.

free variable → vapaa muuttuja 92

function → funktio 67

funktio Laskutoimitus tai mikä tahansa muu jonkin arvon riippuvuus yhdestä tai useammasta arvosta. Esimerkiksi ihmisen syntymävuosi on ihmisen funktio, ja painoindeksi on ihmisen pituuden ja painon funktio. Tässä yhteydessä riippuvuudeksi voidaan katsoa myös näennäinen riippuvuus. Esimerkiksi $0x$ voidaan katsoa x:n funktioksi, vaikka sen arvo ei todellisuudessa riipu x:n arvosta. Myös jopa pelkkä 0 voidaan katsoa x:n funktioksi.

funktiosymboli Katso nimikirjoitus 76.

Gödel numbering → Gödel-numerointi 67

Gödel-numerointi Monenlaisten olioiden, niille tehtävien toimenpiteiden ja niistä esitettävien väitteiden matkiminen luonnollisilla luvuilla, niiden yhteen- ja kertolaskulla sekä niistä puhuvilla kaavoilla.

Gödel-numerointi on teoreettisen tutkimuksen väline. Tarvittavat luvut ovat melkein aina liian suuria ja kaavat liian pitkiä ja vaikeatajuisia käytännöllisiin tarpeisiin. Gödel-numeroinnin yksityiskohdat eivät ole olennaisia, ja ne voivat vaihdella eri kirjoittajilla.

Mikä tahansa äärellinen jono luonnollisia lukuja voidaan pakata yhteen luonnolliseen lukuun keinoilla, jotka perustuvat lukujen jaollisuusoppiin. Siitä seuraa, että Gödel-numeroinnilla voi matkia askel askeleelta eteneviä mielivaltaisen (mutta äärellisen) pitkiä toimintoketjuja. Siksi hyvin monimutkaisia asioita voidaan Gödel-numeroida. Esimerkiksi mille tahansa tietokoneohjelmalle voidaan muodostaa luonnollisten lukujen kaava, joka on tosi tai epätosi sen mukaan tuleeko tietokoneohjelman suoritus tuottamaan tietyn lopputuloksen.

Gödelin epätäydellisyyslauseet ???

Gödelin täydellisyyslause ???

halting tester → pysähtymistesteri, katso 81.

higher order → korkeampi kertaluku, katso ensimmäinen kertaluku 66.

hyvin muodostettu kaava Ilmaus, jolla korostetaan sitä, että tarkoitetaan merkkijonoa joka muodostaa kaavan eikä mitä tahansa merkkijonoa, jossa on samoja merkkejä kuin kaavoissa umpimähkäisessä järjestyksessä. Esimerkiksi $x = 2(y+1)$ on hyvin muodostettu kaava, mutta $y2)x(1 = +$ ei ole. Niinpä ”hyvin muodostettu kaava” tarkoittaa käytännössä samaa kuin ”kaava”.

implication → implikaatio, katso materiaalinen implikaatio 74, looginen seuraus 71 ja päättelyimplikaatio 80.

implikaatio Katso materiaalinen implikaatio 74, looginen seuraus 71 ja päättelyimplikaatio 80.

integer → kokonaisluku 69

interpretation → tulkinta 90

joukko Katso alkio 63.

kaava Katso ensin atomikaava 64. Propositiologiikan kaavoja ovat kaikki atomikaavat sekä ilmaukset muotoa $\neg(kaava)$, $(kaava1) \wedge (kaava2)$, $(kaava1) \vee (kaava2)$, $(kaava1) \rightarrow (kaava2)$ ja $(kaava1) \leftrightarrow (kaava2)$. Predikaattilogiikan kaavoja ovat lisäksi kvanttorikaavat (katso kvanttori 70). Sulkeita ”(” ja ”)” saa jättää pois käytössä olevien suljesääntöjen mukaisesti.

kategorinen Aksiomatisointi on kategorinen, jos ja vain jos se määrittelee puheenaiheen sisäisen rakenteen olennaisesti yksikäsitteisesti, tarkemmin sanoen, jos ja vain jos kaikki aksiomatisoinnin mallit ovat keskenään isomorfiset. Kategorisuus on sikäli täydellisyyttä vahvempi ominaisuus, että se kiinnittää huomiota myös niihin mallien välisiin eroihin, joita ei voi ilmaista kaavoina.

Muun muassa luonnollisille luvuille ja reaaliluvuille on olemassa kategoriset toisen kertaluvun aksiomatisoinnit. Ensimmäisen kertaluvun aksiomatisointi voi olla kategorinen vain siinä tapauksessa, että kaikki mallit ovat äärellisiä (tämä seuraa ylöspäin Löwenheim–Skolemin lauseesta).

kertaluku (logiikassa) Katso ensimmäinen kertaluku 66 ja toinen kertaluku 88.

kieli (logiikassa) Ne äärelliset merkkijonot, jotka esittävät kaavoja (katso atomikaava 64, kaava 68 ja kvanttori 70).

Predikaattilogiikan kieli ei ole yksi eikä edes ensimmäisen kertaluvun kieli ole yksi, vaan jokainen nimikirjoitus (katso nimikirjoitus 76) luo oman kielensä.

Propositiologiikan kieliä voi ajatella olevan vain yksi. Todellisuudessa eri kirjoittajien versiot propositiologiikan kielestä poikkeavat toisistaan esimerkiksi käyttämällä erilaisia suljesääntöjä ja samoille asioille eri symboleita, mutta nämä erot eivät ole syvällisiä.

kokonaisluku Kokonaisluvut ovat $\ldots, -2, -1, 0, 1, 2, \ldots$.

kolmannen poissuljetun laki Jos P on kaava, niin $P \vee \neg P$ on tosi jokaisessa tilanteessa. Tilanteesta riippuen voi olla että P on tosi (jolloin $\neg P$ ei ole) tai $\neg P$ on tosi (jolloin P ei ole), mutta $P \vee \neg P$ on aina tosi.

kompaktisuuslause Numeroituvalla joukolla ensimmäisen kertaluvun suljettuja kaavoja on malli, jos ja vain jos sen jokaisella äärellisellä osajoukolla on malli. (Lause pätee tietyin ehdoin myös ylinumeroituville joukoille.)

Lause seuraa helposti täydellisyyslauseesta: jos joukolla ei ole mallia, niin siitä voidaan päätellä ristiriita. Koska päättelyt ovat äärellisiä, käyttää ristiriidan johtaminen vain äärellistä määrää joukon kaavoja. Ne muodostavat äärellisen osajoukon, jolla ei ole mallia. Vastakkainen suunta on ilmeinen: koko joukon malli on myös sen jokaisen osajoukon malli.

Kompaktisuuslauseesta seuraa muun muassa, että saavutettavuutta suunnatussa graafissa (kuten maantiekartassa, katso saavutettavuus 85) ei voi ilmaista ensimmäisen kertaluvun kaavana. Nimittäin kaavoilla ”mistään solmusta ei lähde useampaa kuin yksi kaari”, ”mistä tahansa solmusta voi saavuttaa minkä tahansa solmun” sekä ”solmuja on ainakin yksi”, ”solmuja on ainakin kaksi”, ”solmuja on ainakin kolme”, … ei ole malleja, mutta niiden millä tahansa äärellisellä osajoukolla on mallina ainakin mikä tahansa silmukka, jossa on ainakin vaadittu määrä solmuja.

konjunktio Symboli ”$\wedge$” tai kaava muotoa *kaava*1 $\wedge$ *kaava*2. Jälkimmäinen on tosi täsmälleen silloin kun sekä *kaava*1 että *kaava*2 on tosi.

korkeampi kertaluku Katso ensimmäinen kertaluku 66.

kunta Joukko olioita, joille on määritelty nolla, ykkönen, yhteenlasku ja kertolasku siten, että ne noudattavat tuttuja laskusääntöjä: yhteenlasku ja kertolasku ovat vaihdannaiset ja liitännäiset, kertolasku noudattaa osittelulakia yhteenlaskun yli (eli sulkeet saa kertoa auki), olio ei muutu kun siihen lisää nollan tai sen kertoo ykkösellä, jokaisella oliolla on vastaolio (olio $+$ vastaolio $= 0$) ja jokaisella muulla oliolla kuin nollalla on käänteisarvo (olio $\cdot$ käänteisarvo $= 1$), ja $0 \neq 1$. Katso myös luku 72.

kvanttori Laajassa käytössä on kaksi eri kvanttoria: *kaikkikvanttori* "$\forall$" ja *olemassaolokvanttori* "$\exists$". Ilmaus $\exists x : kaava$ tarkoittaa, että on olemassa ainakin yksi sellainen arvo, että jos se annetaan x:lle, niin *kaava* on tosi. Esimerkiksi $\exists x : 2 < x < 3$ tarkoittaa, että kakkosen ja kolmosen välissä on ainakin yksi luku. Se on tosi reaaliluvuilla mutta ei kokonaisluvuilla. Ilmaus $\forall x : kaava$ tarkoittaa, että riippumatta siitä, mikä arvo x:lle annetaan, on *kaava* tosi. Esimerkiksi $\forall x : x \leq 2 \vee x \geq 3$ tarkoittaa, että jokainen luku on enintään kaksi tai vähintään kolme. Se on tosi kokonaisluvuilla mutta ei reaaliluvuilla.

Välimerkkien käyttö kvanttori-ilmauksissa vaihtelee eri kirjoittajilla. Siihen kohtaan, jossa tässä tekstissä on kaksoispiste, jotkut laittavat pisteen. Varsinkin teoreettisissa teksteissä on tavallista, että siinä kohdassa ei ole mitään tai on hieman tyhjää. Jos *kaava* on monimutkainen, niin sen ympärille voidaan laittaa sulkeet, koska ilman niitä syntyisi vaikeasti hahmotettavia ilmauksia kuten $\exists x 2 < x < 3$. Jotkut kirjoittavat $\exists x(2 < x < 3)$ ja jotkut $(\exists x)(2 < x < 3)$. Tässä tekstissä ja MathCheckissä kirjoitetaan $\exists x : 2 < x < 3$.

Pääteltäessä on tavallista, että kvanttorilla luodun muuttujan tilalle laitetaan termi (katso termi 87) ja kvanttori poistetaan. Tällekin on käytössä eri merkintätapoja. Tässä tekstissä *termi*:n sijoittaminen x:n jokaisen vapaan esiintymän tilalle *kaava*:ssa ilmaistaan kirjoittamalla $kaava(x)$ ja $kaava(termi)$. Tarvittaessa täytyy lisätä sulkeet. Esimerkiksi $\forall a : 0a = 0$ ja $2x+1$ tuottavat tällä tavalla $0(2x+1) = 0$.

Kun vapaan muuttujan esiintymän tilalle sijoitetaan termi, täytyy varmistaa, että mikään termissä esiintyvä muuttuja ei joudu sidotuksi. Esimerkiksi $\exists y : y > x$ on tosi jokaisella luvulla x. Siihen saa sijoittaa x:n tilalle esimerkiksi $a+1$, jolloin saadaan jokaisella luvulla a tosi kaava $\exists y : y > a+1$. Mutta jos sijoitetaankin $y+1$, niin saadaan aina epätosi kaava $\exists y : y > y+1$. Tämän voi välttää vaihtamalla kvanttorilla luodun muuttujan nimi toiseksi ennen termin sijoittamista. Esimerkiksi voidaan vaihtaa y:n tilalle z, jolloin $\exists y : y > x$ muuttuu muotoon $\exists z : z > x$. Kun siihen sijoitetaan x:n tilalle $y+1$, saadaan $\exists z : z > y+1$. Se on tosi jokaisella luvulla y.

Sama ongelma voi syntyä kun osakaavan tilalle sijoitetaan toinen osakaava. Sen voi ratkaista samalla tavalla.

Matematiikassa on tavallista jättää kaikkikvanttorit kirjoittamatta. Niitä korvaamassa saattaa olla sanallinen ilmaus tai ei mitään. Sekä matematiikassa että logiikassa on tavallista käyttää päättelyissä avoimia kaavoja siten, että kaavan täytyy olla tosi kaikilla mahdollisilla vapaiden muuttujien arvojen yhdistelmillä (katso yhtäpitävyys 92). Se on monimutkaisempi ver-

sio kaikkikvanttorien kirjoittamatta jättämisestä, sillä se vastaa ilmausta $\forall x_1 : \ldots \forall x_n : \neg\, oletukset \vee kaava$.

Kvanttoreille pätevät omat versionsa De Morganin laeista: $\neg\forall x : kaava \Leftrightarrow \exists x : \neg\, kaava$ ja $\neg\exists x : kaava \Leftrightarrow \forall x : \neg\, kaava$.

Jos x ei esiinny vapaana *kaava*:ssa, niin $\forall x : kaava$ ja $\exists x : kaava$ tarkoittavat samaa kuin *kaava*. Jos puheenaiheen joukossa on vain äärellinen määrä eri arvoja ja ne ovat a_1, a_2, …, a_n, niin $\forall x : kaava(x)$ tarkoittaa samaa kuin $kaava(a_1) \wedge kaava(a_2) \wedge \ldots \wedge kaava(a_n)$ ja $\exists x : kaava(x)$ tarkoittaa samaa kuin $kaava(a_1) \vee kaava(a_2) \vee \ldots \vee kaava(a_n)$.

Ilmaukset $\forall x$; *kaava*1 : *kaava*2 ja $\exists x$; *kaava*1 : *kaava*2 tarkoittavat samaa kuin $\forall x : \neg\, kaava1 \vee kaava2$ ja $\exists x : kaava1 \wedge kaava2$ (tarvittaessa sulkeet lisäten). Ilmaukset $\forall x \in A : kaava$ ja $\exists x \in A : kaava$ tarkoittavat samaa kuin $\forall x : x \notin A \vee kaava$ ja $\exists x : x \in A \wedge kaava$ (tarvittaessa sulkeet lisäten).

käytön ja mainitsemisen ero Jos kysymykseen ”mihin talvi loppuu” vastataan ”lumien sulamiseen”, niin sanaa ”talvi” käytettiin. Mutta jos siihen vastataan ”i-kirjaimeen” , niin sana ”talvi” mainittiin. Kun sanaa tai symbolia ”käytetään”, niin sana tai symboli tarkoittaa sitä mitä se tavallisesti tarkoittaa eli edustaa kohdettaan, mutta kun sana tai symboli ”mainitaan”, niin se tarkoittaa itseään tai jotakin itsensä ja varsinaisen kohteensa välissä olevaa asiaa. Jälkimmäisestä esimerkkinä on kun todetaan, että $1+1$ ei ole sama kuin 2, koska $1+1$ on laskutoimitus.

Tässä tekstissä pyritään korostamaan tätä eroa lisäämällä lainausmerkit mainintojen ympärille (vertaa $1+1=2$ ja ”$+$” on vaihdannainen), mutta paikoitellen on koettu selkeämmäksi poiketa tästä käytännöstä esimerkiksi siksi, että muuten lainausmerkkejä tulisi häiritsevän paljon.

language (in logic) → kieli (logiikassa) 68

law of excluded middle → kolmannen poissuljetun laki 69

liitännäinen Laskutoimitus $\circ$ on liitännäinen jos ja vain jos $\forall a : \forall b : \forall c : (a \circ b) \circ c = a \circ (b \circ c)$. Esimerkiksi yhteenlasku ja kertolasku ovat liitännäiset, mutta vähennyslasku ja jakolasku eivät ole. Logiikan ”$\wedge$”, ”$\vee$” ja ”$\leftrightarrow$” ovat liitännäiset, mutta ”$\rightarrow$” ei ole.

logical symbol → looginen symboli, katso ei-looginen symboli 65.

looginen seuraus Kaava on looginen seuraus joukosta kaavoja, jos ja vain jos se on tosi jokaisessa tulkinnassa ja tilanteessa, jossa joukon jokainen kaava on tosi. (Katso tulkinta 90 ja tilanne 88.) Usein käytetään merkintää *joukko* $\models$ *kaava*.

Loogisen seurauksen käsite riippuu todellakin sekä tulkinnasta että tilanteesta. Esimerkiksi $x+2=3$ ei ole kaavan $x=1$ looginen seuraus, vaikka symbolien ”1”, ”2”, ”3” ja ”+” vakiintuneilla tulkinnoilla $x+2=3$ on tosi aina silloin kun $x=1$ on tosi. Ollakseen looginen seuraus sen pitää päteä kaikilla muillakin tulkinnoilla, kuten sillä, jossa ”1” tarkoittaa ”vappu”, ”2” tarkoittaa ”kuusi”, ”3” tarkoittaa ”joulukuusi” ja ”+” tarkoittaa merkkijonojen liittämistä peräkkäin. Kun x:n arvo on ”vappu”, sanoo $x=1$ että vappu = vappu, ja sehän on tosi. Samalla $x+2=3$ sanoo että vappukuusi = joulukuusi, ja se ei ole tosi.

Sen sijaan $x+2=3$ on looginen seuraus, kun joukon kaavat ovat $x=1$ ja $1+2=3$. Edellä ollut vastaesimerkki ei ole enää pätevä, koska $1+2=3$ ei päde sille, koska $1+2$ tarkoittaa ”vappukuusi” mutta 3 tarkoittaa ”joulukuusi”. Jos tulkinta pidetään muuten samana mutta ”1” tarkoittaa ”joulu”, niin $x=1$ on tosi vain kun x:n arvo on ”joulu”. Silloin $x+2=3$ sanoo että joulukuusi = joulukuusi, ja se on tosi.

Käytännön sovelluksissa oletetaan melkein aina, että vakiintuneilla symboleilla on vakiintuneet merkitykset, eikä että ”3” ei välttämättä tarkoitakaan kolmosta vaan saattaakin tarkoittaa vaikka joulukuusta. Siksi loogisen seurauksen käsite ja sitä tarkoittava merkintä ”$\models$” olisivat kömpelöitä logiikan käytännön sovelluksissa. Merkinnän ”$\models$” sijaan käytetään enimmäkseen luonnollista kieltä ja toisinaan symboleita ”$\Rightarrow$” ja ”$\Leftrightarrow$”, katso päättelyimplikaatio 80 ja yhtäpitävyys 92.

Symboleiden vakiintuneet merkitykset voidaan ottaa loogisen seurauksen käsitteessä huomioon ottamalla symboleiden peruslait mukaan siihen joukkoon, jonka loogisesta seurauksesta puhutaan. Matemaattisen päättelyn vaihe ei tyypillisesti ole edellisen vaiheen looginen seuraus, vaan esimerkiksi edellisen vaiheen ja reaalilukujen yleisten lakien looginen seuraus.

looginen symboli Katso ei-looginen symboli 65.

luku Matematiikassa on määritelty useita käsitteitä jotka loppuvat sanaan ”luku”, mutta käsitettä ”luku” pelkkänä ei ole määritelty. Sanaa ”luku” käytetään tyypillisesti tarkoittamaan oliota, joka noudattaa lukujen tavallisia laskusääntöjä eli kunta-aksioomia. Laskusäännön käsite ei viittaa yksittäiseen lukuun, vaan joukkoon lukuja. Esimerkiksi laskussa $1+2=3$ esiintyy kolme eri lukua. Niinpä luku ei ole yksinään mitään, vaan luvusta tekee luvun jäsenyys joukossa, jonka kaikki alkiot yhdessä noudattavat sovittuja laskusääntöjä. Katso kokonaisluku 69, luonnollinen luku 72 ja reaaliluku 82.

luonnollinen luku Luonnolliset luvut ovat $0, 1, 2, \ldots$. Katso tosi aritmetiikka 88.

lyhenne Ilmaus, joka määritelmiä pilkuntarkasti tulkiten ei ole osa kaavojen kieltä, mutta joka edustaa jotakin kaavojen kielen ilmausta ja jota voi käytännössä käyttää ikään kuin se olisi osa kaavojen kieltä. Tyypillisesti lyhenne korvaa ilmauksen lyhyemmällä tai tutummalla. Esimerkiksi $vasen \neq oikea$ on usein käytetty lyhenne ilmaukselle $\neg(vasen = oikea)$ ja $vasen \leq keski \leq oikea$ on usein käytetty lyhenne ilmaukselle $vasen \leq keski \wedge keski \leq oikea$.

Lyhenteiden jättäminen varsinaisen kaavojen kielen ulkopuolelle pienentää sitä, ja siten helpottaa logiikan teoreettista tutkimista. Toisinaan on aivan mielivaltaista, mikä merkintä on lyhenne ja mikä ei ole. Esimerkiksi joissakin kirjoituksissa "$\leq$" on lyhenne mutta "$<$" ei ole, ja joissakin muissa toisinpäin.

Löwenheim–Skolemin lause Katso Skolemin paradoksi 86.

mahdottomasta seuraa mitä tahansa Jokainen "jos ... niin ..." -ilmaus, jonka "jos"-osa ei voi koskaan toteutua, on pätevä. Englanniksi tämä periaate on *principle of explosion*. Se on välitön seuraus siitä, että päättely on pätevä, jos ja vain jos sille ei ole olemassa vastaesimerkkiä eli tilannetta, jossa kaikki lähtökohdat ovat tosia mutta johtopäätös ei ole. Nimittäin jos lähtökohtien toteutuminen on kaiken kaikkiaan mahdotonta, niin mahdotonta on myös, että ne toteutuvat samalla kun johtopäätös ei päde.

Tämä ei välttämättä tarkoita, että johtopäätös olisi tosi. Se tarkoittaa vain, että on oikein päätellä, että jos lähtökohdat olisivat todet, niin johtopäätöskin olisi. Se ei koskaan tuota todellisuuden vastaista johtopäätöstä, koska sitä ei koskaan päästä käyttämään todellisissa tilanteissa, koska lähtökohdat eivät voi toteutua.

Niin sanottu ristiriitatodistus toimii siten, että oletetaan sen vastakohta (tarkemmin sanottuna negaatio) mikä halutaan todistaa, ja päätellään siitä ristiriita. Koska saatiin mahdoton lopputulos, niin jossain täytyy olla jotain vikaa. Jos päättely käytti vain päteviä päättelymenetelmiä, niin vika ei voi olla muualla kuin lähtökohdassa. Niinpä sen, mikä haluttiin todistaa, vastakohta on epätosi, joten se mikä haluttiin todistaa on tosi.

Sen, että mahdottomasta seuraa mitä tahansa, todisti Guillaume de Soissons jo 1100-luvulla. Nykyisin se on ahkerassa käytössä logiikassa.

malli Olkoon annettu joukko suljettuja kaavoja. Sen malli on jokin kokonaisuus, joka toteuttaa joukon jokaisen kaavan. Se voi olla jotakin varsin tuttua kuten kokonaisluvut, tai jotakin hyvin keinotekoista. Katso tarkempia tietoja kohdasta tulkinta 90.

mallin olemassaololause Jokaisella ristiriidattomalla numeroituvalla joukolla ensimmäisen kertaluvun suljettuja kaavoja on malli. (Lause pätee tietyin ehdoin myös ylinumeroituville joukoille.)

Lause voidaan todistaa rakentamalla malli vaiheittain kuvitteellisella (ei välttämättä oikeasti toteutettavissa olevalla) prosessilla, jonka Leon Henkin julkaisi 1949. Idea on yksinkertainen, mutta täsmällinen todistus on pullollaan teknisiä yksityiskohtia. Käydään kaikki kaavat (myös avoimet) järjestyksessä läpi. Kullekin valitaan, onko se tosi, siten, että ei synny ristiriitaa alkuperäisten kaavojen eikä aikaisempien valintojen kanssa. Lisäksi kullekin kaavalle muotoa $\exists x : P$ varataan yksi arvo, joka toteuttaa P:n, ja vastaavasti $\neg\forall x : P$:lle arvo, joka toteuttaa $\neg P$:n. Tätä arvoa kutsutaan *Henkin-todistajaksi* (*Henkin witness*).

Jos alkuperäinen joukko on numeroituva, niin myös edellä kuvatulla tavalla tuotettu malli on numeroituva. Siksi Löwenheim–Skolemin lauseen (katso Skolemin paradoksi 86) vanhin versio seuraa mallin olemassaololauseesta.

materiaalinen ekvivalenssi Symboli ”$\leftrightarrow$” tai kaava muotoa $kaava1 \leftrightarrow kaava2$. Jälkimmäinen on tosi täsmälleen silloin kun joko molemmat tai ei kumpikaan kaavoista $kaava1$ ja $kaava2$ on tosi. Siis $kaava1 \leftrightarrow kaava2 \Leftrightarrow kaava1 \wedge kaava2 \vee \neg kaava1 \wedge \neg kaava2$.

Sana ”materiaalinen” jätetään usein pois, eli puhutaan vain ekvivalenssista.

Symboli ”$\leftrightarrow$” luetaan usein ”jos ja vain jos”. Sillä on samankaltainen suhde symboliin ”$\Leftrightarrow$” kuin symbolilla ”$\rightarrow$” on symboliin ”$\Rightarrow$”, katso materiaalinen implikaatio 74.

Symbolia ”$\leftrightarrow$” kutsutaan englanniksi myös nimellä *biconditional*.

materiaalinen implikaatio Symboli ”$\rightarrow$” tai kaava muotoa $kaava1 \rightarrow kaava2$. Jälkimmäinen on tosi täsmälleen silloin kun $kaava1$ on epätosi tai $kaava2$ on tosi. Siis $kaava1 \rightarrow kaava2 \Leftrightarrow \neg kaava1 \vee kaava2$.

Sana ”materiaalinen” jätetään usein pois, eli puhutaan vain implikaatiosta.

Symboli ”$\rightarrow$” luetaan usein ”jos … niin …”. Toisaalta luonnollisen kielen ”jos … niin …” ei aina tarkoita samaa kuin ”$\rightarrow$”, vaan toisinaan se tarkoittaa päättelysääntöä. Ero ilmenee esimerkiksi virkkeessä ”jos talo on sininen, niin se on vihreä, tai jos se on vihreä, niin se on sininen”. Luonnollisen kielen mukaan virke on väärin, koska kumpikaan sanan ”tai” vastakkaisilla puolilla olevista vaihtoehdoista ei ole oikein, koska talo ei voi olla samanaikaisesti vihreä ja sininen. Mutta kuten seuraavaksi näemme, jos ”jos … niin …” tulkitaan materiaalisena implikaationa, niin virke on aina tosi.

Sinisille taloille ”se on vihreä” on epätosi, joten ”jos se on vihreä, niin se on sininen” on tosi. Muunvärisille taloille ”talo on sininen” on epätosi, joten ”jos talo on sininen, niin se on vihreä” on tosi. Niinpä aina ”tai”:n toinen tai toinen puoli on tosi. Toinen tapa on käyttää yhtäpitävyyttä $kaava1 \rightarrow kaava2 \Leftrightarrow \neg kaava1 \vee kaava2$. Siten virke muuttuu muotoon ”talo ei ole sininen tai se on vihreä, tai se ei ole vihreä tai se on sininen”. Se on selvästi aina tosi. Samankaltainen analyysi paljastaa, että materiaalisen implikaation mukaan jopa ”jos talo on sininen, niin lehmät lentävät, tai jos se on vihreä, niin kuu on vihreää juustoa” on aina tosi.

Lisää tästä aiheesta löytyy Wikipedian sivulta ”Paradoxes of material implication” ja Stanford Encyclopedia of Philosophyn sivulta ”The Logic of Conditionals”. Tässä tekstissä noudatetaan kantaa, jonka mukaan edellä olleissa esimerkeissä jokainen ”jos ... niin ...” pitää tulkita päättelysäännöksi, eli sellaiseksi, että se pätee jokaiselle talolle. Niin tulkittuna ensimmäisen esimerkin virke tarkoittaa, että joko kaikille taloille pätee, että jos se on sininen niin se on vihreä, tai kaikille taloille pätee, että jos se on vihreä niin se on sininen. Kumpikaan vaihtoehto ei toteudu, joten virke on väärin.

Melko monet kirjoittajat käyttävät materiaalisen implikaation symbolina ”$\Rightarrow$” eikä ”$\rightarrow$”. Osa muista kirjoittajista joko ei käytä lainkaan symbolia ”$\Rightarrow$”, tai käyttää sitä ilmaisemaan päättelysääntöä tai päättelyaskelta. Tämä teksti ja MathCheck noudattavat viimeksi mainittua tapaa. Katso päättelyimplikaatio 80.

material equivalence $\rightarrow$ materiaalinen ekvivalenssi 74

material implication $\rightarrow$ materiaalinen implikaatio 74

meta Meta viittaa asian tarkastelemiseen sen itsensä ulkopuolelta. Suurin osa siitä, mitä opiskellaan logiikan nimissä, on itse asiassa metalogiikkaa. Logiikkaa ovat varsinaisesti vain tiukan muodolliset kaavat ja niillä tiukan muodollisesti tapahtuva päättely. Esimerkiksi $kaava \wedge \mathsf{F} \Leftrightarrow \mathsf{F}$ on metalogiikkaa. Logiikaksi se muuttuu vasta sitten, kun kohdassa ”*kaava*” on oikeasti jokin kaava.

model $\rightarrow$ malli 73, katso myös tulkinta 90.

model existence theorem $\rightarrow$ mallin olemassaololause 74

modus ponens Kaavoista $kaava1$ ja $kaava1 \rightarrow kaava2$ saa päätellä $kaava2$. Katso myös deduction theorem 65.

muuttuja Olio, joka saa arvon jostakin ennalta määrätystä joukosta siten, että arvo saattaa vaihdella tilanteesta toiseen. Esimerkiksi ”hän” on luonnollisen

kielen muuttuja, jonka arvo on ihminen. Joissakin tilanteissa ”hän” tarkoittaa George Boolea, joissakin Kurt Gödeliä ja niin edelleen. Jotkin muuttujat saavat arvoikseen reaalilukuja, jotkin kokonaislukuja, jotkin äärellisiä merkkijonoja ja niin edelleen. Katso myös muuttujasymboli 76.

muuttujasymboli Muuttujan nimi. Kaavassa voi olla monta eri muuttujaa joilla on sama nimi joko siten, että kukin erikseen on luotu kvanttorilla, tai yksi on vapaa ja kukin muu on luotu kvanttorilla. ”Muuttuja” ja ”muuttujasymboli” eivät siis ole täsmälleen sama asia. Silti usein sanotaan ”muuttuja” vaikka tarkoitetaan muuttujasymbolia, koska niiden ero on harvoin olennainen ja koska ”muuttujasymboli” on pitkä sana.

natural number → luonnollinen luku 72

negaatio Symboli ”$\neg$” tai kaava muotoa $\neg$*kaava*. Jälkimmäinen on tosi täsmälleen silloin kun *kaava* on epätosi.

negation → negaatio 76

nimikirjoitus Kokoelma ei-loogisia symboleita eli symboleita, jotka edustavat puheenaiheen laskutoimituksia, vertailuja ja niin edelleen. Kukin kokoelman symboli on joko *vakiosymboli*, *funktiosymboli* tai *relaatiosymboli*. Vakiosymbolit edustavat puheenaiheen joukon alkioita (esimerkiksi kokonaislukua 3 tai ihmistä ”Kurt Gödel”), funktiosymbolit edustavat puheenaiheen laskutoimituksia ja funktioita (esimerkiksi yhteenlaskua tai toimintoa, joka saatuaan ihmisen nimen kertoo hänen äitinsä nimen), ja relaatiosymbolit edustavat puheenaiheen alkioista totuusarvoja tuottavia toimintoja eli relaatioita ja kyllä/ei-ominaisuuksia (esimerkiksi ”$\leq$” ja ”syntyi kaupungissa nimeltä Brno”).

Kullakin symbolilla on *paikkaluku*, joka kertoo, kuinka monta arvoa symboli ottaa syötteekseen. Vakiosymbolin paikkaluku on aina nolla ja funktiosymbolin suurempi kuin nolla. Nollapaikkainen funktiosymboli olisi sama asia kuin vakiosymboli, mutta on osoittautunut käteväksi puhua erikseen vakio- ja funktiosymboleista.

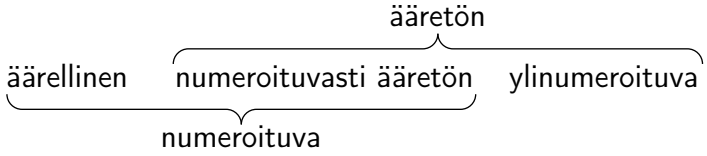
Relaatiosymboleiden paikkaluku voi aina olla suurempi kuin nolla, mutta riippuu kirjoittajasta, voiko se olla nolla. Nollapaikkainen relaatiosymboli olisi melko hyödytön, koska sama vaikutus saadaan selkeämmin symboleilla F ja T. (Sen sijaan voi olla mielekästä ottaa käyttöön propositiomuuttujia. Ne eroavat nollapaikkaisista relaatiosymboleista samalla tavalla kuin tavalliset muuttujat eroavat vakiosymboleista. Ne eivät kuulu nimikirjoitukseen.)

Katso myös tulkinta 90.

non-logical symbol → ei-looginen symboli 65

number → luku 72

numeroituva joukko Äärellinen joukko, tai joukko jonka alkiot voi asettaa yksi–yhteen-vastaavuuteen luonnollisten lukujen kanssa. Numeroituvan joukon vastakohta on *ylinumeroituva* joukko. Esimerkiksi kokonaislukujen, rationaalilukujen ja äärellisestä aakkostosta muodostettujen äärellisten merkkijonojen joukot ovat numeroituvasti äärettömiä, mutta reaalilukujen joukko, äärellisestä aakkostosta muodostettujen päättymättömien merkkijonojen joukko ja luonnollisten lukujen osajoukkojen joukko ovat ylinumeroituvia.



open formula → avoin kaava 64

order → kertaluku, katso ensimmäinen kertaluku 66 ja toinen kertaluku 88.

osajoukko Katso alkio 63.

osittelulaki Sulkeiden auki kertomisen laki. Laskutoimitus $\star$ noudattaa osittelulakia laskutoimituksen $\circ$ yli, jos ja vain jos $\forall a : \forall b : \forall c : a \star (b \circ c) = (a \star b) \circ (a \star c) \wedge (c \circ b) \star a = (c \star a) \circ (b \star a)$. Kertolasku noudattaa osittelulakia yhteen- ja vähennyslaskun yli. Logiikan $\wedge$ noudattaa osittelulakia $\vee$:n yli ja $\vee$ noudattaa osittelulakia $\wedge$:n yli.

paikkaluku Katso nimikirjoitus 76.

Peanon aritmetiikka ???

poissuljetun Katso kolmannen poissuljetun laki 69.

predikaattilogiikka Logiikan laji, jossa käsitellään jotakin puheenaihetta, kuten reaaliluvut tai ihmiset. Valtaosa tästä tekstistä käsittelee predikaattilogiikkaa. Tämä kohta yhdessä ”katso”-kehotuksissa mainittujen kohtien kanssa muodostaa tiivistelmän kaikkein keskeisimmistä asioista.

Monet predikaattilogiikan kaavat sisältävät osia, joita kutsutaan termeiksi. Termi tuottaa arvon puheenaiheen joukosta, siis esimerkiksi reaaliluvun. Termissä voi olla muuttujia kuten x sekä puheenaiheen mukaan määräytyviä

vakiosymboleita kuten 512 ja funktiosymboleita kuten ”+”. Termien tuottamista tuloksista tuotetaan totuusarvoja symbolilla ”=” ja puheenaiheen mukaan määräytyvillä relaatiosymboleilla, kuten ”$\leq$”.

Kaavoista voi muodostaa isompia kaavoja symboleilla ”$\neg$”, ”$\wedge$”, ”$\vee$”, ”$\rightarrow$” ja ”$\leftrightarrow$” kuten propositiologiikassa, sekä kvanttoreilla ”$\forall$” ja ”$\exists$”. Ilmaus $\forall x$: *kaava* on tosi, jos ja vain jos valitaanpa x:lle mikä tahansa arvo niin *kaava* on tosi. Ilmaus $\exists x$: *kaava* on tosi, jos ja vain jos on olemassa yksikin sellainen arvo, että kun x saa sen, niin *kaava* on tosi.

Katso puheenaihe 78, nimikirjoitus 76, atomikaava 64, kaava 68, kvanttori 70, ensimmäinen kertaluku 66 ja toinen kertaluku 88.

principle of explosion Katso mahdottomasta seuraa mitä tahansa 73.

proof $\rightarrow$ todistus, katso päättely 79.

propositiologiikka Logiikan laji, jossa käsitellään pelkästään totuusarvoja eikä erillistä puheenaihetta (kuten reaaliluvut tai ihmiset). Tyypillisesti käytössä ovat propositiomuuttujat, ”$\wedge$”, ”$\vee$”, ”$\neg$” ja ”$\rightarrow$” sekä ehkä myös ”$\leftrightarrow$”. Kvanttoreita, puheenaiheen muuttujia, nimikirjoitusta, termejä ja yhtäsuuruutta ei käytetä.

Kun propositiomuuttujille on annettu totuusarvot, voidaan propositiologiikan kaavan totuusarvo selvittää helposti laskemalla. Myös se onko propositiologiikan kaava aina tosi voidaan periaatteessa aina selvittää laskemalla, mutta käytännössä työmäärä voi olla mahdottoman suuri. Kuuluisa kysymys, onko $\mathbf{P} = \mathbf{NP}$, liittyy tähän. Käytännöllisten keinojen etsiminen, joilla voidaan selvittää monesta isosta propositiologiikan kaavasta, onko se aina tosi, on oma tutkimusalansa.

propositiomuuttuja Muuttuja, jonka arvo on totuusarvo.

propositional logic $\rightarrow$ propositiologiikka 78

propositional variable $\rightarrow$ propositiomuuttuja 78

puheenaihe Se, mistä kaavat puhuvat. Esimerkiksi kokonaisluvut, reaaliluvut, äärelliset merkkijonot tai ihmiset. Sanaa käytetään kahdella eri tavalla. Kapean käyttötavan mukaan ”puheenaihe” tarkoittaa muuttujille mahdollisia arvoja, siis esimerkiksi kaikkia kokonaislukuja tai kaikkia ihmisiä. Lavean käyttötavan mukaan ”puheenaihe” kattaa lisäksi arvojen käsittelyyn, vertaamiseen ja niin edelleen käytettävät symbolit, kuten ”+” ja ”$\leq$” (katso nimikirjoitus 76), sekä niiden ominaisuudet, kuten $\forall a : a \leq a + 1$.

Puheenaiheen joukko tarkoittaa puheenaihetta kapeasti tulkittuna. Puheenaiheen joukossa on oltava ainakin yksi alkio.

pysähtymistesteri Katso 81.

päättely Matemaattisessa logiikassa päättely tarkoittaa kaavojen tuottamista aikaisemmista kaavoista tai tyhjästä ennalta sovitun *päättelyjärjestelmän* (*proof system*) tarjoamilla keinoilla. (On järkevää, että esimerkiksi kaavan $x = x$ voi tuottaa tyhjästä.) Päättelyn lähtökohtana on joukko puheenaiheen ominaisuuksia ilmaisevia suljettuja kaavoja (katso aksiooma 62), sekä mahdollisesti tapauskohtaisia oletuksia kuten ”jos $x \geq 0$, niin ...”. Päättelyjärjestelmä ei riipu puheenaiheesta.

Kun on valittu jokin päättelyjärjestelmä, niin merkintä *kaavoja* $\vdash$ *kaava* tarkoittaa, että mainitun kaavan voi tuottaa mainitusta joukosta kaavoja.

Matemaattisen logiikan päättelyjärjestelmissä päättely muodostuu pienistä askelista. Se on eduksi sen tutkimiselle mitä voidaan ja mitä ei voida päätellä, mutta tekee päättelyjärjestelmistä kovin kömpelöitä käytännön päättelytyöhön. Esimerkiksi peruskoulussa kaksi samanmuotoista termiä voidaan yhdistää yhtenä askeleena, mutta tyypillisessä päättelyjärjestelmässä sen tekeminen edellyttää useaa vaihetta, joissa käytetään reaalilukujen osittelulakia, ykkösen määritelmää ja niin edelleen.

Päättelyjärjestelmiltä vaaditaan, että kukin päättely koostuu äärellisestä määrästä askelia (jotta se valmistuisi joskus) ja että ainakin periaatteessa on olemassa tietokoneohjelma, joka pystyy tarkastamaan kunkin askeleen (jotta päättelyn pätevyys ei jäisi mielipidekysymykseksi; rekursiivisesti lueteltavuus riittää, katso rekursiivinen 83). Tästä seuraa, että voidaan laatia tietokoneohjelma, joka luettelee jokaisen kaavan jonka päättelyjärjestelmä tuottaa ja vain ne. Tällainen ohjelma olisi kuitenkin melkein aina aivan liian hidas käytännön tarpeisiin.

Vaihtoehtoisia päättelyjärjestelmiä on monia erilaisia. Tyypillisessä järjestelmässä on yksi tai useampia *päättelysääntöjä* ja nolla tai useampia *loogisia aksioomia*. Päättelysäännöt kertovat, miten käytettävissä olevista kaavoista saa tuottaa (lisää) kaavoja. Loogiset aksioomat ovat kaavoja, jotka eivät kerro puheenaiheen ominaisuuksia vaan logiikan sisäisiä tosiasioita. Puheenaiheen ominaisuuksia kertovat aksioomat ovat *ei-loogisia* aksioomia. Loogisten aksioomien ja päättelysääntöjen ero ei ole olennainen tämän tekstin kannalta. Loogiset aksioomat mainitaan tässä vain siksi, että muuta kirjallisuutta lukiessa on hyvä tietää, että sana ”aksiooma” saattaa tarkoittaa muutakin kuin ei-loogisia aksioomia.

Päättelyjärjestelmä on *totuuden säilyttävä* (*sound*), jos ja vain jos sillä ei voi päätellä tosista lähtökohdista epätosia johtopäätöksiä. Päättelyjärjestelmä on *täydellinen* (*complete*), jos ja vain jos sillä voi päätellä jokaisen kaavan, joka on tosi aina silloin kun kaikki lähtökohdat ovat todet (katso tulkinta 90). Ensimmäisen kertaluvun logiikalle on ja toisen kertaluvun logiikalle ei ole olemassa täydellisiä totuuden säilyttäviä päättelyjärjestelmiä. Päättelyjärjestelmän täydellisyyttä ei pidä sekoittaa aksioomajärjestelmän täydellisyyteen, katso aksiooma 62.

Päättelyjärjestelmistä puhuttaessa *todistus* tarkoittaa kaavan tuottamista totuuden säilyttävässä päättelyjärjestelmässä. Yleisemmin logiikassa ja matematiikassa todistus tarkoittaa epämuodollista päättelyä, jota aiheeseen perehtyneet matemaatikot pitävät pätevänä. Jokainen pätevä matemaattinen todistus voidaan periaatteessa esittää jossakin päättelyjärjestelmässä, mutta käytännössä niin tehdään vain harvoin. Päättelyn esittäminen päättelyjärjestelmässä ja tarkastaminen tietokoneella ei yleensä antaisi täyttä varmuutta virheettömyydestä, koska tarvitaan epätäsmällistä päättelyä alkuperäisen ongelman kääntämiseksi kaavojen kielelle.

Jos päättelyjärjestelmä on totuuden säilyttävä ja täydellinen, ja jos jonkin suljetun kaavan totuus määräytyy lähtökohdista, niin edellä mainittu tuotettavissa olevat kaavat luetteleva ohjelma selvittää onko se tosi, löytämällä todistuksen joko sille itselleen tai sen negaatiolle. Jos lähtökohdat jättävät suljetun kaavan totuuden auki tai päättelyjärjestelmä on epätäydellinen, niin voi olla mahdotonta havaita, että sitä ei voi todistaa todeksi eikä epätodeksi.

päättelyimplikaatio Symboli ”$\Rightarrow$” tai ilmaus muotoa *kaava*1 $\Rightarrow$ *kaava*2. Tässä tekstissä ja MathCheckissä jälkimmäinen tarkoittaa, että jokaisessa mahdollisessa tilanteessa, jossa *kaava*1 on tosi, myös *kaava*2 on tosi (mutta ei välttämättä päinvastoin). Mahdollisen tilanteen käsitteestä katso yhtäpitävyys 92. Symbolien ”$\Rightarrow$” ja ”$\rightarrow$” välillä vallitsee samanlainen suhde kuin symbolien ”$\Leftrightarrow$” ja ”$\leftrightarrow$”, katso yhtäpitävyys 92.

*Kaava*1 $\Leftrightarrow$ *kaava*2 pätee jos ja vain jos sekä *kaava*1 $\Rightarrow$ *kaava*2 että *kaava*2 $\Rightarrow$ *kaava*1 pätee. Jos *kaava*1 $\Rightarrow$ *kaava*2 ja *kaava*2 $\Rightarrow$ *kaava*3 niin *kaava*1 $\Rightarrow$ *kaava*3, mutta ei välttämättä toisinpäin. Pääsääntöisesti *kaava*1 $\Rightarrow$ *kaava*2 ei anna lupaa korvata isomman kaavan osana olevaa *kaava*1 kaavalla *kaava*2.

Kenties yleisin ”$\Rightarrow$”:n käyttötapa tämän tekstin ja MathCheckin ulkopuolella on materiaalisen implikaation symbolina, siis siinä tehtävässä, missä tämä teksti, MathCheck ja osa logiikan kirjallisuutta käyttää symbolia ”$\rightarrow$” (katso materiaalinen implikaatio 74). Symbolia ”$\Rightarrow$” käytetään kirjallisuudessa jonkin verran myös samankaltaisesti kuin MathCheckissä, mutta täsmentämättä sen merkitystä.

ratkeamattomuus *Pysähtymistesteriksi* kutsutaan tietokoneohjelmaa, joka mistä tahansa tietokoneohjelmasta ja sille annetusta syötteestä selvittää, tuleeko se pysähtymään mainitulla syötteellä. Pysähtymistestereitä ei voi olla olemassa. Jos olisi, niin olisi helppo kirjoittaa ohjelma, joka pysähtymistesteriä käyttäen ennustaa, tuleeko se itse pysähtymään kun se saa oman itsensä syötteekseen, ja vastauksen saatuaan tekee päinvastoin. On muitakin tapoja todistaa, että pysähtymistesteri on mahdottomuus.

Siis pysähtymisen testaaminen on ratkeamaton tehtävä. Tämä ei tarkoita, että koskaan ei voitaisi saada selville, tuleeko annettu ohjelma pysähtymään annetulla syötteellä. Se tarkoittaa vain, että ei ole olemassa yleispätevää menetelmää, joka pystyisi jokaisesta ohjelmasta ja sen syötteestä selvittämään, tuleeko se pysähtymään. On olemassa epätäydellisiä pysähtymistestereitä, jotka toisinaan vastaavat ”en tiedä” tai jäävät laskemaan ikuisesti tuottamatta mitään vastausta.

Jokainen epätäydellinen pysähtymistesteri epäonnistuu äärettömälle määrälle tapauksia. Nimittäin jos se epäonnistuisi vain äärelliselle määrälle, voitaisiin se muuttaa täydelliseksi lisäämällä eteen vaihe, jossa se tunnistaisi nämä erikoistapaukset ja katsoisi niille oikeat vastaukset etukäteen laaditusta luettelosta.

Mille tahansa tietokoneohjelmalle ja sen syötteelle voidaan kirjoittaa luonnollisista luvuista puhuva suljettu kaava, joka on tosi jos ja vain jos ohjelma pysähtyy mainitulla syötteellä (katso Gödel-numerointi 67). Siinä kaavassa ei ole muita ei-loogisia symboleita kuin yhteenlasku, kertolasku ja lukuvakioita kuten 0 ja 1. On olemassa ohjelma, joka tuottaa sen automaattisesti, kun sille annetaan ohjelma ja sen syöte.

Jos olisi olemassa ohjelma, joka mistä tahansa luonnollisten lukujen suljetusta kaavasta mainituilla symboleilla selvittää, onko se tosi, niin kytkemällä se edellä mainitun kaavantuottajaohjelman perään saataisiin pysähtymistesteri. Joten koska pysähtymistestereitä ei ole olemassa, ei ole olemassa myöskään ohjelmaa, joka selvittäisi mistä tahansa luonnollisten lukujen suljetusta kaavasta yhteen- ja kertolaskuilla sekä lukuvakioilla, onko se tosi.

Kuuluisat matemaatikot David Hilbert ja Wilhelm Ackermann asettivat 1928 tavoitteeksi tällaisen ohjelman löytämisen. Tosin koska silloin ei vielä ollut tietokoneita eikä niiden ohjelmia, he eivät puhuneet ohjelmasta vaan ”efektiivisestä menetelmästä”, mutta sittemmin on tultu siihen tulokseen, että ne ovat sama asia. Tavoite tunnettiin nimellä *Entscheidungsproblem*.

Alonzo Church todisti 1936, että Entscheidungsproblem on ratkeamaton. Hänen todistuksessaan ”efektiivinen menetelmä” oli rajattu kapealta vaikuttavalla tavalla, joten jäljelle jäi mahdollisuus, että olisi olemassa jokin hänen

todistuksensa ulkopuolelle jäävä keino sittenkin ratkaista se. Myöhemmin samana vuonna Alan Turing julkaisi tutkimuksen, jossa hän esitteli teoreettisen tietokoneen ja osoitti sen suoriutuvan hyvin monenlaisista tehtävistä. Sitä käyttäen hänkin osoitti, että Entscheidungsproblem on ratkeamaton.

Turingin koneeseen perustuva efektiivisyyden määritelmä oli hyvin vakuuttava. Sittemmin osoitettiin, että vaikka se on hyvin erilainen kuin Churchin määritelmä, se kuitenkin johtaa täsmälleen samaan rajaukseen sille mikä on ja mikä ei ole ratkeavaa. Näistä tapahtumista ja niihin liittyvistä käsitteistä katso rekursiivinen 83.

Sittemmin lukuisat muut laskenta- ja päättelytehtävät on osoitettu ratkeamattomiksi. Ratkeamattomuus on hyvin yleinen ilmiö.

Jotta päättelymenetelmä voisi olla käytännössä käyttökelpoinen, on oltava olemassa tietokoneohjelma, joka pystyy tarkastamaan, onko esitetty päättely pätevä. Tätäkin on käsitelty tarkemmin kohdassa rekursiivinen 83. (Riittää, että pätevien päättelyiden joukko on rekursiivisesti lueteltava.)

Ensimmäisen kertaluvun logiikalle on olemassa rekursiivisia päättelymenetelmiä, jotka pystyvät todistamaan kaikki suljetut kaavat, jotka ovat tosia kaikissa niissä tulkinnoissa, joissa kaikki aksioomat ovat tosia. Sellaiselle päättelymenetelmälle voidaan kirjoittaa ohjelma, joka tutkii suljetun kaavan totuutta kokeilemalla päättelyitä järjestelmällisesti kunnes löytää kaavalle todistuksen. Siis jokaiselle ensimmäistä kertalukua olevalle aksiomatisoinnille pätee, että sen puitteissa todistettavien suljettujen kaavojen joukko on rekursiivisesti lueteltava.

Koska luonnollisten lukujen yhteen- ja kertolaskulla suljettujen kaavojen totuuden selvittäminen on ratkeamatonta, täytyy mille tahansa niiden aksiomatisoinnille olla olemassa suljettuja kaavoja, joiden totuuden aksiomatisointi jättää avoimeksi. Muussa tapauksessa minkä tahansa suljetun kaavan totuus voitaisiin selvittää kokeilemalla päättelyitä kunnes löytyy todistus joko kaavalle itselleen tai sen negaatiolle.

reaaliluku Reaalilukuja ovat kokonaisluvut ja kokonaislukujen välissä olevat luvut. Reaalilukujen joukko on suurin lukujoukko, jossa kaikki luvut asettuvat suuruusjärjestykseen.

reaalisuljettu kunta Mikä tahansa kunta, joka toteuttaa täsmälleen samat ensimmäisen kertaluvun suljetut kaavat kuin reaaliluvut toteuttavat, kun eiloogiset symbolit ovat ”0”, ”1”, ”$+$”, ”$\cdot$” ja ”$\leq$”. Se voidaan määritellä muuten samoin kuin reaaliluvut tavallisesti määritellään, mutta niin sanotun täydellisyysaksiooman sijaan vaaditaan, että jokaisella positiivisella re-

aaliluvulla on neliöjuuri ja jokaisella paritonta astetta olevalla polynomilla on nollakohta (katso täydellisyysaksiooma 91).

Reaalisuljettujen kuntien teoria on siinä mielessä täydellinen, että jokainen ensimmäisen kertaluvun suljettu kaava edellä mainituilla symboleilla voidaan todistaa todeksi tai epätodeksi. Silti on olemassa sekä reaalilukuja pienempiä että reaalilukuja suurempia reaalisuljettuja kuntia. Ne voidaan erottaa reaaliluvuista toisen kertaluvun kaavoilla. Tietenkin reaaliluvut mainituilla symboleilla ovat itse reaalisuljettu kunta. Muut reaalisuljetut kunnat kuin reaaliluvut itse voidaan ajatella reaalilukujen epästandardeiksi malleiksi. Katso myös täydellisyysaksiooma 91.

reachability → saavutettavuus 85

real closed field → reaalisuljettu kunta 82

real number → reaaliluku 82

recursive → rekursiivinen 83

rekursiivinen Logiikassa sanalla ”rekursiivinen” tarkoitetaan tietokoneohjelmalla laskettavissa olevaa. Esimerkiksi joukko äärellisiä merkkijonoja on rekursiivinen, jos ja vain jos ainakin periaatteessa on olemassa tietokoneohjelma, joka mille tahansa äärelliselle merkkijonolle vastaa äärellisessä ajassa, kuuluuko se kyseessä olevaan joukkoon vai eikö kuulu. Tässä yhteydessä ”joukko” on olennaisesti sama asia kuin kyllä/ei-kysymys. Esimerkiksi kolmella jaollisten kokonaislukujen joukko vastaa kysymystä ”onko syötteeksi annettu luku kolmella jaollinen”.

Joukko on *rekursiivisesti lueteltava*, jos ja vain jos ainakin periaatteessa on olemassa tietokoneohjelma, joka pysähtyy joukon alkioille ja laskee ikuisesti muille syötteille. Se siis vastaa ”kyllä” pysähtymällä ja ”ei” laskemalla loputtoman kauan. Rekursiivinen lueteltavuus on epäsymmetrinen: kun ohjelma on pysähtynyt, on varmaa että vastaus on ”kyllä”, mutta jos ohjelma ei vielä ole pysähtynyt, niin vastaus saattaa olla kumpi tahansa. Saattaahan olla, että ohjelma pysähtyisi, jos jaksettaisiin odottaa vielä vähän aikaa, mutta saattaa olla myös, että se ei koskaan pysähdy. Kyllä/ei-kysymys on rekursiivinen jos ja vain jos se itse on rekursiivisesti lueteltava, ja myös se mikä saadaan vaihtamalla ”kyllä” ja ”ei” keskenään on rekursiivisesti lueteltava.

On olemassa monia hyvin määriteltyjä logiikan tai muusta näkökulmasta tärkeitä joukkoja, jotka eivät ole rekursiivisesti lueteltavia. Myös on tärkeitä joukkoja, jotka ovat rekursiivisesti lueteltavia mutta eivät rekursiivisia. Katso ratkeamattomuus 81.

Logiikassa käytetään myös käsitettä ”primitive recursive”. Se ei tarkoita täsmälleen samaa kuin rekursiivinen, mutta ero on niin pieni ja sellaiseen suuntaan, että siitä tarvitsee harvoin välittää. Jokainen ”primitive recursive” joukko tai funktio on rekursiivinen, mutta ei toisinpäin.

Logiikassa ei tietenkään voida hyväksyä päättelyaskeleen perusteeksi ”syö kaksi kärpässientä, niin näet seuraavana yönä unessa, että tämä päättelyaskel on pätevä” eikä ”tämä päättelyaskel on pätevä, koska kansakuntamme suuri ja iäti kunnioitettu kaikkitietävä johtaja sanoi niin”. Siksi haluttiin, että päättelyaskelien on oltava objektiivisesti tarkastettavissa. Alkuun ei kuitenkaan osattu antaa matemaattiseen työskentelyyn sopivaa määritelmää sille, mitä ”objektiivisesti tarkastettavissa” tarkkaan ottaen tarkoittaa. Niinpä käyttöön otettiin sana *efektiivinen* (englanniksi *effective*) sitä kuvaamaan, kunnes kunnollinen määritelmä keksittäisiin.

1900-luvun alkupuolella tehtiin useita yrityksiä antaa tarkka määritelmä efektiivisyydelle. Niiden joukossa oli muuten lupaavia, mutta ne vaikuttivat kapea-alaisilta. Ei ollut hyvää syytä uskoa, että ne kattaisivat kaikki objektiivisen tarkastamisen muodot.

Tilanne muuttui täysin 1936. Silloin Alan Turing julkaisi kirjoituksen, jossa hän esitteli sittemmin kuuluisaksi tulleen teoreettisen koneensa ja osoitti, että sillä voidaan laskea ja päätellä hyvin monenlaisia asioita. On jopa olemassa niin sanottu *universaali Turingin kone*, joka voi matkia kaikkia Turingin koneita. Universaali Turingin kone ja sille annettu matkittavan Turingin koneen kuvaus ovat 1930-luvun teoreettinen versio siitä, mitä nykyisin kutsutaan tietokoneeksi ja ohjelmaksi. ”Suoritettavissa Turingin koneella” oli vakuuttava, laaja-alainen määritelmä efektiivisyydelle. Nykyisin saman voi sanoa ”laskettavissa tietokoneohjelmalla”.

Mikä vielä parempaa, pian huomattiin, että monet aikaisemmat (ja myöhemmät) yritykset määritellä efektiivisyys rajasivat sen täsmälleen samoin kuin Turingin koneet, vaikka olivat teknisesti hyvin erilaisia. Niinpä efektiivisyyden käsitteelle olikin vain yksi vakavasti otettava vaihtoehto, ja kilpailevat määritelmät olivat vain erilaisia tapoja ilmaista tämä vaihtoehto. *Church–Turing teesi* tarkoittaa väitettä, että oikea efektiivisyyden käsite on tämä yhteinen käsite.

Matematiikassa (ja ohjelmoinnissa) ”rekursiivinen” tarkoittaa itseään kutsuvaa tai itseään apunaan käyttävää. Yksi efektiivisyyden vaihtoehtoisista määritelmistä perustui tässä mielessä rekursiivisiin funktioihin. Tätä perua ”rekursiivinen” päätyi tarkoittamaan logiikassa edellä mainittua yhteistä efektiivisyyden käsitettä, joka voidaan ilmaista ”laskettavissa tietokoneohjelmalla”.

relaatio Mikä tahansa, joka ottaa kaksi tai useampia arvoja ja tuottaa totuusarvon. Esimerkiksi ”$<$” ottaa kaksi lukua ja tuottaa ”tosi” jos ja vain jos ensimmäinen on pienempi kuin jälkimmäinen. ”Sisarus” ottaa kaksi ihmistä ja tuottaa ”tosi” jos ja vain jos heillä on samat vanhemmat.

relaatiosymboli Katso nimikirjoitus 76.

relation → relaatio 85

ristiriidaton Joukko kaavoja on ristiriidaton, jos ja vain jos siitä ei voi päätellä sekä jotakin kaavaa että sen negaatiota, eli siitä ei voida päätellä F. Ristiriidattomuus on päättelemiseen liittyvä käsite, ja sitä vastaava kaavojen merkitykseen liittyvä käsite on tyydytettävyys 91.

ristiriita Jokin väite voidaan päätellä sekä todeksi että epätodeksi. Toisin sanoen, voidaan päätellä F.

saavutettavuus *Suunnattu graafi* (*directed graph*) on rakenne, jossa on kahdenlaisia osia: solmuja ja kaaria. Solmut piirretään ympyröinä, ja kukin kaari piirretään nuolena ympyrästä ympyrään. Solmut voivat edustaa esimerkiksi maantiekartan risteyksiä ja kaaret risteyksien välisiä yksisuuntaisia tienpätkiä. Kaksisuuntainen tienpätkä saadaan laittamalla kaari kumpaankin suuntaan.

Solmu on saavutettavissa solmusta, jos ja vain jos ensin mainitusta on reitti jälkimmäiseen. Reitti tarkoittaa nollaa tai useampaa kaarta siten, että seuraava kaari alkaa siitä solmusta mihin edellinen kaari päättyy. Esimerkiksi Suomen tiekartassa Utsjoki on saavutettavissa Turusta, mutta Maarianhamina ei ole, koska Maarianhamina on saaressa, johon ei ole siltoja eikä tunneleita mantereelta.

satisfiable → tyydytettävissä 91

schema Katso aksioomaskeema 63.

second order → toinen kertaluku 88

sentence → suljettu kaava 87

shorthand → lyhenne 73

sidottu muuttuja Muuttuja, joka on luotu kvanttorilla. Sidottu muuttuja käy läpi kaikki puheenaiheen joukon alkiot.

signature → nimikirjoitus 76

Skolemin paradoksi Se tosiasia, että ylinumeroituvien joukkojen olemassaolo voidaan todistaa joukko-opin ensimmäisen kertaluvun numeroituvassa aksiomatisoinnissa, vaikka jokaisella ensimmäisen kertaluvun numeroituvalla aksiomatisoinnilla, jolla ylipäänsä on malleja, on numeroituva malli. Miten sellaiseen malliin, jossa kaiken kaikkiaan on vain numeroituvasti ääretön määrä alkioita, voi mahtua joukko, jossa on ylinumeroituvasti alkioita?

Leopold Löwenheim todisti 1915, että jos ensimmäisen kertaluvun suljetulla kaavalla on malli, niin sillä on numeroituva malli. Vuonna 1922 Thoralf Skolem yleisti tuloksen numeroituviin ensimmäisen kertaluvun suljettujen kaavojen joukkoihin. Tätä tulosta kutsutaan *Löwenheim–Skolemin lauseeksi*. Skolem nosti paradoksin esiin samassa julkaisussa, ja antoi sille aika hyvän selityksen. Nykyisin ollaan hyvin laajasti sitä mieltä, että Skolemin paradoksi ei ole matematiikan näkökulmasta ongelma.

(Löwenheim–Skolemin lauseesta on sittemmin kehitetty kaksi kaikista erisuurista äärettömyyksistä puhuvaa lausetta, joita kutsutaan alaspäin ja ylöspäin Löwenheim–Skolemin lauseiksi. Katso myös mallin olemassaololause 74.)

Joukko on numeroituva jos ja vain jos sen alkiot voidaan asettaa yksi–yhteen-vastaavuuteen luonnollisten lukujen kanssa. Yksi–yhteen-vastaavuus on itsessään joukko. Skolemin paradoksin selitys lähtee siitä tosiasiasta, että joukko-opin aksiomatisoinnilla on useita eri malleja. Joukko voi olla yhdessä mallissa ylinumeroituva ja toisessa numeroituva siksi, että jälkimmäisessä on ja edellisessä ei ole ainakin yksi yksi–yhteen-vastaavuus, joka numeroi sen. Ylinumeroituvuus on *suhteellinen* (englanniksi *relative*) eikä *absoluuttinen* (*absolute*) ominaisuus, eli se riippuu mallista.

Niinpä voidaan ajatella, että jokainen joukko-opin malli olisi siten vajaa, että siitä puuttuu joukkoja. Ylinumeroituva joukko olisikin ”ulkopuolelta katsottuna” numeroituva, mutta malli ei sitä huomaa, koska siitä puuttuvat ne joukot, jotka numeroisivat sen. Ei olisi mikään ihme, että joukko-opin aksiomatisointi olisi tällä tavalla puutteellinen, sillä se on ensimmäistä kertalukua, ja ensimmäinen kertaluku on monella muullakin tavalla puutteellinen.

Skolemin paradoksi ei siis ole matemaattinen ristiriita. Silti se on filosofisesti kiinnostava, ja siitä on paljon kirjallisuutta.

sound → totuuden säilyttävä 89

standard model → standardimalli 87

standardimalli Malli, joka täsmää puheenaiheen joukon ja nimikirjoituksen symboleiden vakiintuneisiin merkityksiin. Esimerkiksi luonnollisten lukujen standardimalli sisältää vain luvut $0, 1, 2, \ldots$ ja niiden yhteen- ja kertolasku toimivat kuten alakoulussa opetettiin, mutta epästandardit mallit sisältävät ylimääräisiä lukuja (katso tosi aritmetiikka 88).

structure → struktuuri 87

struktuuri Joukko-opin keinoin muodostettu järjestelmä, jota vasten kaavan totuusarvo määräytyy. Struktuuri koostuu epätyhjästä joukosta, jota tässä tekstissä kutsutaan puheenaiheen joukoksi, sekä joukko-opillisesta edustajasta jokaiselle nimikirjoituksen symbolille. Katso nimikirjoitus 76.

suljettu kaava Kaava, jossa ei esiinny yhtään vapaata muuttuja. Kun nimikirjoituksen tulkinta on kiinnitetty, tuottaa suljettu kaava joko aina totuusarvon ”tosi” tai aina totuusarvon ”epätosi”.

symbol → symboli 87

symboli Jonkin asian merkki tai nimi. Katso ei-looginen symboli 65 (myös looginen symboli), nimikirjoitus 76 (vakiosymboli, funktiosymboli ja relaatiosymboli) sekä muuttujasymboli 76.

Tennenbaumin lause Katso tosi aritmetiikka 88.

teoria Katso aksiooma 62.

term → termi 87

termi Predikaattilogiikan kielen alarakenne, joka määräytyy nimikirjoituksen mukaan. Termi tuottaa arvon puheenaiheen joukosta. Esimerkiksi $2x$ ja $2x+3$ ovat termejä.

Yksittäinen vakiosymboli kuten 3 on termi, ja yksittäinen muuttuja kuten x on termi. Myös laskutoimitus tai funktiosymboli, jolle on annettu syötteeksi paikkaluvun mukainen määrä termejä, on termi. Funktiosymboli voi olla peräisin nimikirjoituksesta, tai korkeamman kertaluvun tapauksessa se voi olla luotu kvanttorilla.

Termien kirjoitustapa riippuu puheenaiheesta. On tavallista käyttää sulkeita ”(” ja ”)” ohjaamaan termin rakennetta. Esimerkiksi $2(x+3)$ on eri termi kuin $2x+3$, mutta $(2x)+3$ on sama termi kuin $2x+3$.

theory → teoria, katso aksiooma 62.

tilanne Vapaiden muuttujien arvojen yhdistelmä. ”Tilanne” ei ole tämän käsitteen vakiintunut suomenkielinen nimi, vaan käytössä on muun muassa ”tulkintajono”. ”Tilanne” valittiin tähän tekstiin, koska sen merkitys esimerkiksi ilmauksessa ”jääkiekko-ottelun tilanne on 5–3” vastaa erittäin hyvin käsitettä ”muuttujien arvojen yhdistelmä”.

Propositiomuuttujien totuusarvojen yhdistelmää voidaan kutsua tulkinnaksi tai tilanteeksi, katso tulkinta 90.

todistus Katso päättely 79.

toinen kertaluku Katso ensin ensimmäinen kertaluku 66. Toisen kertaluvun logiikassa voidaan käyttää kaikkea mitä ensimmäisessä kertaluvussakin, mutta lisäksi kvanttoreilla voidaan luoda muuttujia, jotka saavat arvoikseen puheenaiheen kyllä/ei-ominaisuuksia tai relaatioita. Kvanttoreilla voidaan luoda myös muuttujia, jotka saavat arvoikseen puheenaiheen funktioita tai laskutoimituksia.

Toisen kertaluvun logiikan ilmaisuvoima riittää hyvin suureen osaan siitä, mitä matematiikassa halutaan ilmaista. Esimerkiksi luonnolliset luvut ja reaaliluvut voidaan määritellä kattavasti toisen kertaluvun kaavoilla. Mitalin toinen puoli on, että toisen kertaluvun logiikalle ei ole olemassa päättelyjärjestelmää, jolla voidaan päätellä kaikki todet kaavat ja vain ne. Niinpä toinen kertaluku ei ratkaise esimerkiksi luonnollisten lukujen määrittelemiseen liittyviä ongelmia kokonaan, vaan vain siirtää ne toiseen paikkaan, nimittäin päättelemiseen.

Lisäksi toisella kertaluvulla voidaan kirjoittaa suljettuja kaavoja, joiden totuus riippuu syvällisistä äärettömiä joukkoja koskevista kysymyksistä. Siksi kuuluisa filosofi W. V. O. Quine kutsui toisen kertaluvun logiikkaa ”joukkoopiksi lammasten vaatteissa” [Philosophy of Logic, 1970].

tosi aritmetiikka Luonnolliset luvut 0, 1, 2, … ja niiden yhteen- ja kertolasku sellaisina, kuin ne alakoulussa opittiin. Luonnollisten lukujen ja niiden yhteen- ja kertolaskun ominaisuuksia on mahdotonta määritellä kattavasti (katso Gödelin 1. epätäydellisyyslause 67). Siksi jokaisella luonnollisten lukujen ensimmäisen kertaluvun aksiomatisoinnilla (katso aksiooma 62) on tosi aritmetiikan lisäksi loputtomasti muitakin malleja (katso malli 73). Niissä on ylimääräisiä lukuja. Ne voidaan mieltää äärettömän suuriksi.

Luonnolliset luvut yhteen- ja kertolaskulla voidaan aksiomatisoida toisen kertaluvun logiikalla siten, että jäljelle ei jää muita malleja kuin tosi aritmetiikka. Mutta toisella kertaluvulla ei ole totuuden säilyttävää päättelyjärjestelmää, jossa voidaan päätellä jokainen aksioomien seuraus. Siksi tälläkään tavalla ei saada kiinni luonnollisten lukujen kaikkia ominaisuuksia.

Luonnollisten lukujen yleisimmin käytetty ensimmäisen kertaluvun aksiomatisointi on Peanon aritmetiikka, katso Peanon aritmetiikka 77. Vaikka sekään ei määrittele luonnollisia lukuja kattavasti, se pääsee aika lähelle. *Tennenbaumin lause* (Stanley Tennenbaum 1959) sanoo, että Peanon aritmetiikalla ei ole muita numeroituvia malleja, joissa yhteenlasku ja kertolasku ovat tietokoneella laskettavissa, kuin tosi aritmetiikka.

Esimerkkejä luonnollisten lukujen ominaisuuksista, joita Peanon aritmetiikka ei todista, löytyy Paris–Harringtonin lauseesta (1977) ja Kirby–Parisin lauseesta (1982) (ei tässä tekstissä).

totuuden säilyttävä Päättelyjärjestelmä on totuuden säilyttävä, jos ja vain jos sen piirissä ei voi päätellä tosista lähtökohdista epätosia johtopäätöksiä. Totuuden säilyttävyys on niin yleinen vaatimus logiikassa, että se saatetaan jättää mainitsematta, koska sitä pidetään itsestään selvänä.

totuuden määrittelemättömyyslause Puheenaiheena on luonnolliset luvut yhteen- ja kertolaskulla. Alfred Tarski todisti 1933, että vaikka suljettujen kaavojen joukko voidaan esittää kaavana, tosien suljettujen kaavojen joukkoa ei voida. Kurt Gödel oli näyttänyt 1931, miten kaavoja voi esittää luonnollisten lukujen ja niiden ominaisuuksien avulla (katso Gödel-numerointi 67). Sitä hyödyntäen, ja olettaen että tosien suljettujen kaavojen joukko voitaisiin esittää, Tarski rakensi kaavan, joka sanoo ”minä olen epätosi suljettu kaava”. Mutta ”minä olen epätosi suljettu kaava” on mahdottomuus, koska se ei voi olla totta eikä epätotta.

Tarskin todistus voidaan esittää helppotajuisemmassa muodossa käyttämällä luonnollisten lukujen sijaan merkkijonoja ja niiden peräkkäin liittämistä, sekä W. V. O. Quinen 1962 esittämää muunnelmaa virkkeestä ”minä olen epätosi”:

> ”Laitettuna lainausmerkeissä itsensä eteen tuottaa epätoden”
> laitettuna lainausmerkeissä itsensä eteen tuottaa epätoden.

Jonkin verran joudutaan näkemään vaivaa jotta lainausmerkkejä voisi olla myös lainausten sisällä, mutta vain vähän verrattuna Gödel-numerointiin.

totuus Totuus on kaiken kaikkiaan syvällisiä pohdintoja herättävä käsite, mutta logiikassa sille on vakiintunut Alfred Tarskin 1933 esittämä ja 1956 parantama määritelmä. Se on pohjimmiltaan hyvin yksinkertainen, jopa niin yksinkertainen, että se saattaa tuntua pelkältä jankutukselta joka ei sano mitään tärkeää. Silti se on toimiva, ja sen keksiminen oli tärkeä kehitysaskel.

Jokaiselle loogiselle symbolille asetetaan ilmaus seuraavaan tyyliin:

- $\neg$ *kaava* on tosi, jos ja vain jos *kaava* on epätosi.
- *kaava*1 $\wedge$ *kaava*2 on tosi, jos ja vain jos *kaava*1 on tosi ja *kaava*2 on tosi.
- $\exists x$: *kaava* on tosi, jos ja vain jos on olemassa ainakin yksi x:n arvo, jolla *kaava* on tosi.
- *termi*1 $=$ *termi*2 on tosi jos ja vain jos *termi*1 ja *termi*2 tuottavat saman arvon.

Termit käsitellään tyyliin ”termi *termi*1 $+$ *termi*2 tuottaa sen arvon, joka saadaan laskemalla termien *termi*1 ja *termi*2 tuottamat arvot yhteen”. Katso myös nimikirjoitus 76, tulkinta 90, tilanne 88 ja käytön ja mainitsemisen ero 71.

true arithmetic $\rightarrow$ tosi aritmetiikka 88

truth $\rightarrow$ totuus 89

tulkinta Kun nimikirjoitus on valittu, niin ensimmäisen kertaluvun logiikassa tulkinta saadaan valitsemalla jokin epätyhjä joukko puheenaiheen joukoksi, valitsemalla jokaiselle vakiosymbolille jokin puheenaiheen joukon alkio, valitsemalla jokaiselle funktiosymbolille jokin puheenaiheen laskutoimitus tai funktio, ja valitsemalla jokaiselle relaatiosymbolille jokin puheenaiheen vertailuoperaatio tai muu sellainen. Tulkinnan täytyy täsmätä symbolien paikkalukuihin. Tulkinta voi, mutta sen ei tarvitse, täsmätä symbolien vakiintuneisiin käyttötapoihin.

Tulkinta ei anna arvoja vapaille muuttujille. Sitä varten on eri käsite, katso tilanne 88. Tulkinnan ja tilanteen erottelu on hyödyllinen muun muassa siksi, että tyypillissä logiikan käyttökohteissa tulkinta säilyy mutta tilanne vaihtelee. Esimerkiksi jos käsketään ratkaisemaan yhtälö $x+2=3$, niin symboleilla ”$+$”, ”2” ja ”3” on kiinteät, jo alakoulussa opitut merkitykset, ja ainoa missä on valinnanvaraa on x:n arvo.

Kun on annettu joukko suljettuja kaavoja, niin malli tarkoittaa tulkintaa, joka toteuttaa kaikki sen kaavat.

Propositiologiikassa tulkinta (tai tilanne) saadaan valitsemalla jokaiselle propositiomuuttujalle totuusarvo. Voidaan esittää eri näkemyksiä siitä, pitäisikö tätä kutsua tulkinnaksi vai tilanteeksi. Nimitystä ”tulkinta” puoltaa se, että propositiologiikassakin voi esittää aksiomatisointeja kaavojen joukkoina, ja predikaattilogiikassa aksiomatisointi esittää tulkinnan eikä tilanteen ominaisuuksia. Nimitystä ”tilanne” puoltaa se, että predikaattilogiikassa vapaat muuttujat saavat arvonsa tilanteesta eikä tulkinnasta, ja propositiomuuttujia kutsutaan usein sanalla jonka loppuosa on ”muuttuja”. Tässä

erossa on kyse kuitenkin vain siitä, mikä olisi paras sanavalinta. Erolla ei ole matemaattista merkitystä.

Se, että kaava on tosi jossakin tulkinnassa ja tilanteessa, merkitään usein *tulkinta*, *tilanne* $\models$ *kaava*. Suljetun kaavan tapauksessa *tilanne* on tarpeeton ja voidaan jättää pois. *Tilanne* voi sijaita myös "$\models$":n alaindeksinä.

Turing machine $\rightarrow$ Turingin kone, katso rekursiivinen 83

Turingin kone Katso rekursiivinen 83.

tyydytettävissä Joukko kaavoja on tyydytettävissä, jos ja vain jos sillä on ainakin yksi malli ja tilanne, eli sellaiset ei-loogisten symbolien tulkinta ja vapaiden muuttujien arvojen yhdistelmä, että joukon jokainen kaava on tosi. Tyydytettävyys on kaavojen merkitykseen liittyvä käsite, ja sitä vastaava päättelemiseen liittyvä käsite on ristiriidattomuus 85.

täydellinen aksiomatisointi / teoria Katso aksiooma 62.

täydellinen päättelyjärjestelmä Katso päättely 79.

täydellisyysaksiooma Reaalilukujen määrittelemisessä laajalti käytetty aksiooma, jolla ei ole mitään tekemistä logiikan täydellisyyskäsitteiden kanssa. Se sanoo, että jos reaaliluvut jaetaan millä tahansa tavalla kahteen epätyhjään joukkoon "pienet" ja "suuret" siten, että jokainen pieni reaaliluku on pienempi kuin mikään suuri ja yhdessä ne kattavat kaikki reaaliluvut, niin joko pienten joukossa on kaikkia muita pieniä suurempi tai suurten joukossa on kaikkia muita suuria pienempi. Se varmistaa muun muassa, että neliöjuuri kaksi on mukana reaaliluvuissa. Tätä täydellisyyskäsitettä kutsutaan toisinaan keksijänsä mukaan Dedekind-täydellisyydeksi, jotta se erottuisi logiikan täydellisyyskäsitteistä.

Reaalilukujen täydellisyysaksiooma on toista kertalukua, koska se puhuu reaalilukujen osajoukoista. Se voitaisiin korvata äärettömällä aksioomaskeemalla, joka puhuu osajoukkojen sijaan ensimmäisen kertaluvun kaavoilla määriteltävissä olevista osajoukoista. Siten reaaliluvuille saataisiin ensimmäisen kertaluvun aksiomatisointi. Niin ei kuitenkaan yleensä tehdä, vaan sen sijaan asetetaan aksiooma ja aksioomaskeema sanomaan, että jokaisella positiivisella reaaliluvulla on neliöjuuri ja jokaisella paritonta astetta olevalla polynomilla on nollakohta. Niin saadaan reaalisuljettu kunta, katso reaalisuljettu kunta 82.

Täydellisyysaksioomaan perustuva reaalilukujen teoria on todistettavuuden näkökulmasta epätäydellinen, koska toisen kertaluvun logiikalla ei ole täydellisiä päättelyjärjestelmiä. Toisaalta Tarski ja Seidenberg todistivat, että

reaalisuljettujen kuntien teoria — siis se, jossa ei käytetä täydellisyysaksioomaa — on täydellinen. Tämä kenties paradoksaaliselta vaikuttava tilanne on mahdollinen, koska ensimmäisen kertaluvun ilmaisuvoima on heikompi kuin toisen kertaluvun. Ensimmäisellä kertaluvulla ei voida kirjoittaa kaavaa, joka poimii luonnolliset luvut reaaliluvuista. Toisella kertaluvulla voidaan, jolloin Gödelin 1. epätäydellisyyslauseen oletukset astuvat voimaan.

uncountable set → ylinumeroituva joukko, katso numeroituva joukko 77.

undecidability → ratkeamattomuus 81

undefinability theorem, Tarski's → totuuden määrittelemättömyyslause 89

use–mention distinction → käytön ja mainitsemisen ero 71

vaihdannainen Laskutoimitus $\circ$ on vaihdannainen jos ja vain jos $\forall a : \forall b : a \circ b = b \circ a$. Esimerkiksi yhteenlasku ja kertolasku ovat vaihdannaiset, mutta vähennyslasku ja jakolasku eivät ole. Logiikan "$\wedge$", "$\vee$" ja "$\leftrightarrow$" ovat vaihdannaiset, mutta "$\rightarrow$" ei ole.

vakiosymboli Katso nimikirjoitus 76.

vapaa muuttuja Muuttuja, joka ei ole luotu kvanttorilla. Vapaa muuttuja voi saada eri arvoja eri tilanteissa, katso tilanne 88.

variable → muuttuja 75

vastaesimerkki Mikä tahansa yksittäistapaus, joka on vastoin esitettyä väitettä. Vapaiden muuttujien arvojen yhdistelmä, jolla kaava ei ole tosi, on vastaesimerkki. Myös vapaiden muuttujien arvojen yhdistelmä, jolla yhtäpitävyys tai muu sellainen ei päde, on vastaesimerkki.

well-formed formula → hyvin muodostettu kaava 68

yhtäpitävyys Symboli "$\Leftrightarrow$" tai ilmaus muotoa *kaava*1 $\Leftrightarrow$ *kaava*2. Tässä tekstissä ja MathCheckissä jälkimmäinen tarkoittaa, että jokaisessa mahdollisessa tilanteessa, jossa *kaava*1 on tosi, myös *kaava*2 on tosi, ja päinvastoin. Tilanne tarkoittaa vapaiden muuttujien tai propositiomuuttujien arvojen yhdistelmää. Tilanne on mahdollinen, jos ja vain jos se toteuttaa kaikki puheenaiheen lainalaisuudet (jotka voi olla ilmoitettu esimerkiksi aksioomilla) ja voimassa olevat tilapäiset oletukset. Tilapäisiä oletuksia tehdään esimerkiksi jaettaessa päättelyä tapauksiin tyyliin "jos $x \geq 0$, niin ..." ja "muussa tapauksessa ...". Lisätietoa katso tilanne 88 ja päättely 79.

Toisin kuin "$\leftrightarrow$", symboli "$\Leftrightarrow$" ei sijaitse kaavan sisällä vaan kahden kaavan välissä. Niinpä "$\Leftrightarrow$" ei saa sijaita sulkeiden sisällä. Esimerkiksi sekä $(\mathsf{F} \leftrightarrow \mathsf{T}) \leftrightarrow \mathsf{F}$ että $\mathsf{F} \leftrightarrow \mathsf{T} \leftrightarrow \mathsf{F}$ tuottaa T, mutta $(\mathsf{F} \Leftrightarrow \mathsf{T}) \Leftrightarrow \mathsf{F}$ ei tarkoita mitään, ja $\mathsf{F} \Leftrightarrow \mathsf{T} \Leftrightarrow \mathsf{F}$ on epäpätevä päättely.

*Kaava*1 $\Leftrightarrow$ *kaava*2 pätee jos ja vain jos sekä *kaava*1 $\Rightarrow$ *kaava*2 että *kaava*2 $\Rightarrow$ *kaava*1 pätee. Yhtäpitävyys noudattaa ekvivalenssin lakeja, katso ekvivalenssi 66. Kaavan osan saa korvata yhtäpitävällä kaavalla ja lopputulokset ovat keskenään yhtäpitävät, jos osakaava ja sen tilalle tuleva kaava eivät tuota määrittelemätöntä totuusarvoa eivätkä sisällä sellaisia vapaita muuttujia, joiden arvoja koskien on voimassa oletuksia ja jotka tulevat sidotuiksi koko kaavan osina.

Kenties yleisin "$\Leftrightarrow$":n käyttötapa tämän tekstin ja MathCheckin ulkopuolella on materiaalisen ekvivalenssin symbolina, siis siinä tehtävässä, missä tämä teksti, MathCheck ja osa logiikan kirjallisuutta käyttää symbolia "$\leftrightarrow$" (katso materiaalinen ekvivalenssi 74). Symbolia "$\Leftrightarrow$" käytetään kirjallisuudessa jonkin verran myös samankaltaisesti kuin MathCheckissä, mutta täsmentämättä sen merkitystä.

MathCheckissä "$\Leftrightarrow$" samaistaa epätoden ja määrittelemättömän. Esimerkiksi $\frac{1}{x} \geq 1 \Leftrightarrow 0 < x \leq 1$ on MathCheckissä pätevä, mutta $\frac{1}{x} \geq 1 \Leftrightarrow 0 \leq x \leq 1$ ei ole. Niitä tapauksia varten joissa epätotta ja määrittelemätöntä ei haluta samaistaa katso yhtätotuus 93.

yhtätotuus Symboli "$\equiv$" on matematiikassa ja logiikassa käytössä useassa eri merkityksessä. Usein sillä tarkoitetaan jotakin yhtä suurta tai suurempaa samankaltaisuutta kuin mitä "$=$" ja "$\Leftrightarrow$" tarkoittavat. Tässä tekstissä ja MathCheckissä sitä käytetään varsinkin puhuttaessa propositiomuuttujalle annettavasta tai kaavan tuottamasta totuusarvosta, tai kun jokin symboli otetaan käyttöön tarkoittamaan jotakin kaavaa.

MathCheckin tapauksessa tarkempi selitys on seuraava. Koska nollalla ei voi jakaa, ei kumpikaan kaavoista $\frac{1}{0} \geq 0$ ja $\frac{1}{0} < 0$ ole tosi. Toisaalta molempien julistaminen epätodeksi aiheuttaisi ongelman, koska tavanomaisten sääntöjen mukaan $a < b \Leftrightarrow \neg(a \geq b)$, joten jos $\frac{1}{0} \geq 0$ on epätosi, niin $\frac{1}{0} < 0 \Leftrightarrow \neg(\frac{1}{0} \geq 0) \Leftrightarrow \neg\mathsf{F} \Leftrightarrow \mathsf{T}$.

Tavanomaisessa predikaattilogiikassa tämä vältetään siten, että jakolaskua ei ole käytettävissä. Merkintä $\frac{a}{b}$ voidaan korvata kirjoittamalla sen sijaan c ja lisäämällä kaavan eteen "$\exists c : a = bc \wedge$". Arkimatematiikassa määrittelemättömät lausekkeet käsitellään epämuodollisella päättelyllä.

MathCheckissä on käytössä kolmas totuusarvo "määrittelemätön" niitä tilanteita varten, joissa kaava ei voi olla tosi eikä epätosi. Esimerkiksi yhtälöä

ja sen ratkaisun lopputulosta verrattaessa on olennaista vain, millä vapaiden muuttujien arvoilla kaava on tosi ja millä ei ole. Siksi MathCheckissä *kaava*1 $\Leftrightarrow$ *kaava*2 hyväksyy sen, että *kaava*1 on epätosi ja *kaava*2 määrittelemätön tai toisinpäin tai molemmat ovat määrittelemättömiä.

On kuitenkin tapauksia, joissa epätotta ja määrittelemätöntä ei saa samaistaa. Niitä varten MathCheckissä *kaava*1 $\equiv$ *kaava*2 tarkoittaa, että *kaava*1 ja *kaava*2 tuottavat saman totuusarvon. Jos jompikumpi on määrittelemätön, niin toisenkin pitää olla. Tässä merkityksessä käytettyinä symbolia ”$\equiv$” ja ilmausta muotoa *kaava*1 $\equiv$ *kaava*2 kutsutaan yhtätotuudeksi.

Kuten ”$\Leftrightarrow$” ja ”$\Rightarrow$”, myöskään ”$\equiv$” ei MathCheckissä sijaitse kaavan sisällä vaan kahden kaavan välissä. Jos *kaava*1 $\equiv$ *kaava*2 niin *kaava*1 $\Leftrightarrow$ *kaava*2, mutta ei välttämättä toisinpäin. Yhtätotuus noudattaa ekvivalenssin lakeja, katso ekvivalenssi 66. Kaavan osan saa korvata yhtätodella kaavalla ja lopputulokset ovat keskenään yhtätodet, jos osakaava ja sen tilalle tuleva kaava eivät sisällä sellaisia vapaita muuttujia, joiden arvoja koskien on voimassa oletuksia ja jotka tulevat sidotuiksi koko kaavan osina.

ylinumeroituva joukko Katso numeroituva joukko 77.