Näitä "totuuslaskutoimituksia" kutsutaan *päättelysäännöiksi (inference rules)*.

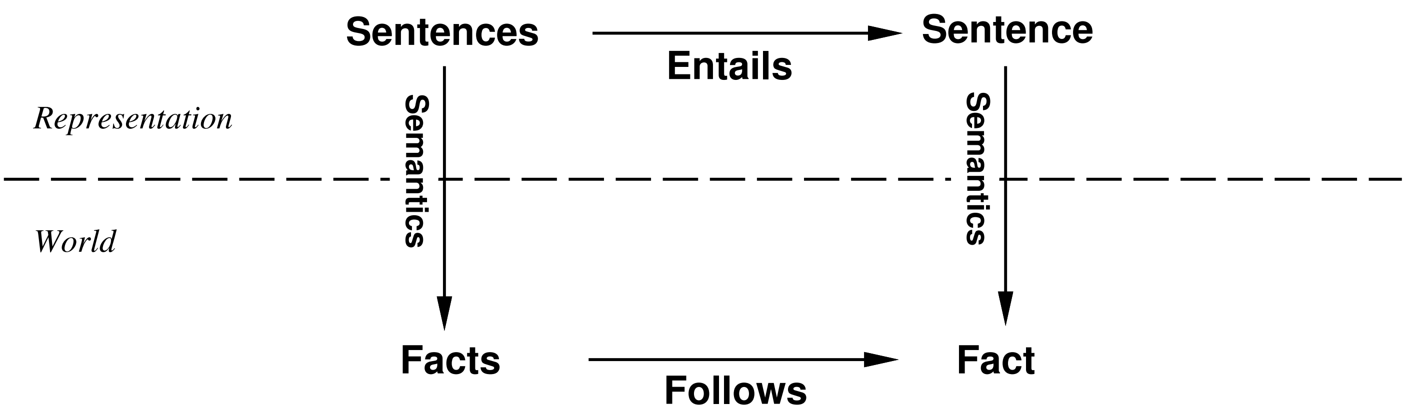
Tarkastellaan tutun Aristotelisen syllogismin (1)–(3) "Scheme-koodausta"

```
(ihminen sokrates)
(jos (ihminen (? . x)) (kuolevainen (? . x)))
---------------------------------------------
(kuolevainen sokrates)
```

jossa tietokone on ohjelmoitu seuraavasti:

"Jos muistissa on ilmaukset A ja (jos B C), ja jos ilmaukset A ja B saadaan samoiksi valitsemalla sopivat arvot joillekin niiden aukkopaikoille (? . y), niin tuo muistiin myös ilmauksen C kopio, jossa nämä aukkopaikat on täytetty samoilla valinnoilla."

Tämä operaatio *tapahtuu kokonaan fysikaalisia tietorakenteita manipuloimalla* mutta *yhtäpitävästi vastaavien "jos. . . ,niin. . . " -lauseiden merkityksen kanssa*:



Kuva 55: Päättely ja seuraus [RN95, kuva 6.5].

Kuva 55 esittää tilanteen kaksijakoisuutta:

- Maailmassa on faktoja, joista *seuraa* toisia faktoja, sen mukaan "miten maailma toimii".

  Esimerkissä faktasta "Sokrates on ihminen" seuraa fakta "Sokrates on kuolevainen".
- Toimijan muistissa on faktojen fysikaalisia *esityksiä (representations)* joista *on pääteltävissä (entails)* päättelysäännöillä toisia

esityksiä.

Esitysten ja päättelysääntöjen tulisi olla symbolisia (ehkä puutteellisia) vastineita sille, miten maailma toimii, jotta tulokset olisivat maailmassa toimijalle hyödyksi.

Esimerkissä esityksestä `(ihminen sokrates)` lasketaan esitys `(kuolevainen sokrates)`.

Tällaisia kieliä tarkastellaan esimerkiksi seuraavin käsittein:

Oikeellisuus (Soundness): Jos päättelysääntö(joukko) tai sen tietokonetoteutus $\textsc{Ask}$ tuottaa tuloksenaan *vain* sellaisia tietämyksenesityskielen lauseita, joiden tulkinnat seuraavat tietämyskannan KB tulkinnasta (= sen sisältämien lauseiden tulkinnoista), niin se on *oikeellinen (sound, truth-preserving)*.

Tämä minimivaatimus on sama asia kuin että metakielen käsitteen "totuus" semantiikkaa ei rikota.

Todistus (Proof): Lista (tai puu tai muu esitys) oikeellisen päättelymenetelmän annetulla syötteellä suorittamista askelista.

Täydellisyys (Completeness), 1. merkitys: Jos *päättelysääntöjoukko* tai sen tietokonetoteutus $\textsc{Ask}$ pystyy tuottamaan tuloksenaan *kaikki* nuo seurauslauseet, niin se on *täydellinen (complete)*.

Erityisesti Gödel osoitti (luku 1.2) ettei luonnollisten lukujen aritmetiikkaa voi koskaan kokonaan vangita minkäänlaiseen mekaaniseen sääntöjärjestelmään.

Täydellisyys, 2. merkitys: *Lausejoukko tai tietämyskanta* KB on täydellinen, jos siitä pystytään päättelemään kaikki nuo seurauslauseet, kun käytetään jotakin täydellistä päättelysääntöjoukkoa.

Täydellisyys on toivottavaa, mutta ei aina mahdollista; sehän vaatii, että todellisuus (tai sen tarkasteltu osa) voitiin muotoilla *tyhjentävästi* säännöiksi.

Täydellisyys, luvun 3 merkitys: Päättelysäännöt kertovat vain sen, millaisia laskutoimituksia voi yleensä käyttää todistusten tekemiseen, ei sitä miten niitä pitäisi käyttää nykyisessä todistuksessa.

Syntyy *todistuksen etsintäongelma.* Siinä täydellisyys tarkoittaa siis:

Jos tarkasteltavalla lauseella *on* jokin todistus, niin tämä etsintämenetelmä *löytää* sen.

Lisäksi toivotaan (muttei aina saavuteta): vaikka ei, niin menetelmä pysähtyy silti.

Monotonisuus (monotonicity): Päättelyjärjestelmä on *monotoninen (monotonic)* jos uusien lauseiden lisääminen tietämyskantaan KB ei tee sen aikaisemman sisällön pohjalta vedettyjä johtopäätöksiä vääriksi.

Silloin tieto "kasvaa".

Seuraava on esimerkki epämonotonisesta päättelystä:

1. Tipi on lintu.
2. Tavalliset linnut lentävät.
3. *Koska minulla ei ole mitään syytä epäillä Tipiä epätavalliseksi*, niin päättelen sen lentävän.

Jos myöhemmin saammekin tietää Tipin olevan pingviini, täytyy meidän silloin perua johtopäätöksemme 3.

Lause- ja predikaattilogiikka ovat monotonisia.

Niissä johtopäätös 3 voidaan tehdä vasta sitten kun Tipin (epä)tavallisuus on ensin varmistettu ja säännön 2 käyttö näin oikeutettu. Ennen sitä kysyjälle vastataan vain "ei tietoa", jottei oikeellisuus kärsisi.

Toteutuvuus (satisfiability): Annettu tietämyksenesityskielen lause ϕ on *toteutuva (satisfiable)* jos *on olemassa* jokin tulkinta jossa lause ϕ on *tosi*.

Kumoutuvuus (falsifiability): Annettu tietämyksenesityskielen lause ϕ on *kumoutuva (falsifiable)* jos *on olemassa* jokin tulkinta jossa lause ϕ onkin *epätosi*.

Tautologisuus (tautologicity, validity, analyticity): Annettu tietämyksenesityskielen lause ϕ on *tautologia (tautology)* jos se *ei ole kumoutuva*.

Nämä lauseet ovat tosia suoraan nykyisen kielen semantiikan ominaisuuksien perusteella.

Esimerkiksi "Sokrates on kuolevainen *tai* Sokrates *ei* ole kuolevainen" on aina totta sen perusteella miten sanojen "tai" ja "ei" kompositionaalinen semantiikka on määritelty käyttämässämme kielessä: "A tai B" on totta jos ainakin toinen väitteistä A tai B on totta, ja "ei C" on totta jos C on epätotta.

On kuitenkin toisia mahdollisia semantiikkoja, jossa näin ei ole, esimerkiksi semantiikka, jossa on käsitteiden *tosi* ja *epätosi* lisäksi vielä kolmaskin *totuusarvo (truth value)*, vaikkapa *"en tiedä"*. Mutta tämä onkin jo eri kieli!

Ristiriitaisuus (unsatisfiability): Annettu tietämyksenesityskielen lause ϕ on *ristiriitainen (unsatisfiable)* jos se *ei ole toteutuva*.

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

Taulukko 3: Kielten ontologisia sitoumuksia [RN95, kuva 6.5].

Tietämyksenesityskieliä voidaan vertailla myös sen suhteen, minkälaisia *ontologisia/epistemologisia sitoumuksia (ontological/epistemological commitments)* niiden määrittelyissä on tehty. (Nämä käsitteet esiteltiin jo luvussa 1.3.) Taulukko 3 vertailee (kohta) tuttujen kielten sitoumuksia.

Luvussa 5.1 esiteltävä (ja jo luvussa 1.3 mainittu) *lauselogiikka (Propositional Logic/Calculus)* on sitoutunut mallintamaan olemassa olevan maailman kokoelmalla esimerkiksi kielioppisäännön (14) mukaisia *jakamattomia* väitelauseita kuten `"sokrates on kuolevainen"`.

Luvussa 5.2 esiteltävä (ja jo luvussa 1.3 mainittu) *[ensimmäisen kertaluvun] predikaattilogiikka ([First Order] Predicate Logic/Calculus)* on taas sitoutunut mallintamaan maailman tällaisten väitelauseiden *osina*: `"Sokrates"` on *olio (object)*, `"kuolevainen"` on *joukko eli yksipaikkainen relaatio (relation)*, ja koko väitelause on niiden yhdistelmä "$\text{Sokrates} \in \text{kuolevainen}$".

5.1. Lauselogiikka

Lauselogiikan syntaksi määritellään antamalla sen *lauseenmuodostussäännöt* induktiivisesti:

Totuusarvovakiot $\mathbf{T}$ ja $\mathbf{F}$ ovat lauseita.

Lausemuuttuja P_i on lause.

Kaikkien lausemuuttujien joukkoa merkitään

$$\mathcal{P} = \{P_i \colon i \in \mathbb{N}\} \,.$$

Joukko on ääretön, jotta muuttujat eivät loppuisi kesken.

Jokainen kaava on äärellinen, ja sisältää siis vain äärellisen monta näistä muuttujista.

Negaatio $(\neg\phi)$ on lause, jos ϕ on.

Konjunktio $(\phi \wedge \psi)$ on lause, jos sekä ϕ että ψ ovat.

Muita lauseita ei lauselogiikassa ole.

Säästetään kirjoitustyötä ottamalla käyttöön. . .

NIMI	MERKINTÄ	TARKOITTAA
Disjunktio	$(\phi \vee \psi)$	$(\neg((\neg\phi) \wedge (\neg\psi)))$
Implikaatio	$(\phi \Rightarrow \psi)$	$((\neg\phi) \vee \psi)$
Ekvivalenssi	$(\phi \Leftrightarrow \psi)$	$((\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi))$

Taulukko 4: Lauselogiikan tavalliset lyhenteet.

ja sopimalla, että sulkuja voi jättää pois laskevana *sidontajärjestyksenä (precedence order)* ilmausten esittelyjärjestys yllä.

Tällöin esimerkiksi $\phi \vee \psi \wedge \chi$ on näkymättömästi sulutettu $(\phi \vee (\psi \wedge \chi))$.

Lauselogiikan semantiikassa aloitetaan *totuusarvojakelusta (truth value distribution)*

$$\mathcal{T} : \mathcal{P} \mapsto \{\mathbf{1}, \mathbf{0}\}$$

missä $\mathbf{1}$ ja $\mathbf{0}$ ovat metakielemme *totuusarvot (truth values)* "tosi" ja "epätosi".

Tämä $\mathcal{T}$ laajennetaan lausemuuttujista kaikkiin lauseisiin samalla induktiolla kuin lauseet määriteltiin (kompositionaalisen semantiikan idea):

Totuusarvovakioiden tulkinnat ovat aina $\mathcal{T}(\mathbf{T}) = \mathbf{1}$ ja $\mathcal{T}(\mathbf{F}) = \mathbf{0}$.

Lausemuuttujien tulkinnat antaa $\mathcal{T}$ suoraan.

Lausemuuttuja on symboli, jonka sovitaan merkitsevän jotakin *kokonaista väitelausetta* — siis sellaista ilmausta Ψ, että "väite Ψ on (epä)tosi" on ymmärrettävää metakieltämme.

Esimerkiksi lausemuuttujan P_1 voitaisiin sopia merkitsevän väitelausetta `"tietokone ajattelee"`, koska "väite 'tietokone ajattelee' on tosi" on ymmärrettävää suomenkieltä.

Sen osa `"tietokone"` ei enää käy lausemuuttujan merkitykseksi, koska "väite 'tietokone' on tosi" ei enää ole ymmärrettävissä.

Erilaiset totuusarvojakelut $\mathcal{T}$ esittävät siis erilaisia käsityksiä tällaisista "peruslauseista".

- Jos tietämyskannan lausemuuttuja P_2 viittaa havainnoitavaan seikkaan kuten "edessä olevan auton jarruvalot palavat", niin $\mathcal{T}(P_2)$ kertoo onko vastaava havainto saatu eli palavatko ne vaiko eivät.

  Nämä lausemuuttujat saavat arvonsa havainnoista tai havaintosarjoissa ilmenneistä säännönmukaisuuksista (kuten "edessä oleva auto vilkuttaa oikealle").
- Jos P_3 viittaa toimijan sisäiseen seikkaan kuten "minun on jarrutettava", niin $\mathcal{T}(P_3)$ kertoo onko seikka "tiedostettu" eli onko minun jarrutettava vai ei.

  Nämä lausemuuttujat saavat arvonsa tietämyskannan säännjöistä havaintomuuttujien arvojen pohjalta: esimerkiksi sääntö "jos P_2

niin P_3" on tarpeen, jotta toimija voisi valita seuraavaksi teokseen jarruttamisen.

Negaatioiden ja konjunktioiden tulkinnat saadaan määräämällä ensin induktiivisesti niiden osien totuusarvot ja käyttämällä sitten vastaavaa *totuustaulua (truth table)*:

$\mathcal{T}(\phi)$	$\mathcal{T}((\neg\phi))$
0	**1**
1	**0**

$\mathcal{T}(\phi)$	$\mathcal{T}(\psi)$	$\mathcal{T}((\phi \wedge \psi))$
0	**0**	**0**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

Taulukko 5: Negaation ja konjunktion totuustaulut.

Näin "$(\neg\phi)$" voidaan lukea "ei $\mathcal{T}(\phi)$", ja "$(\phi \wedge \psi)$" taas "$\mathcal{T}(\phi)$ ja $\mathcal{T}(\psi)$".

Taulukon 4 lyhenteiden totuustaulut voidaan sitten laskea käyttäen taulukkoa 5:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Taulukko 6: Lyhenteiden totuustaulut [RN95, kuva 6.9].

Implikaatio $P \Rightarrow Q$ edustaa muotoa "jos P niin Q" olevia väitteitä.

Nämä väitteet on jo nähty tärkeiksi johtopäätöksiä tehtäessä.

Implikaation sarake taulukossa 6 näyttää, että implikaatio on epätosi vain silloin kun sen *etujäsen (antecedent)* P on tosi mutta *takajäsen (consequent)* Q epätosi.

Yksi lukutapa lauseelle $P \Rightarrow Q$ onkin väite

$$\mathcal{T}(P) \leq \mathcal{T}(Q) \tag{16}$$

missä on sovittu, että $\mathbf{0} < \mathbf{1}$.

Tämä implikaatio *ei siis edusta syy-seuraussuhdetta* ”P aiheuttaa että Q” (vaan on ns. *materiaalinen* implikaatio).

Jokaiseen (lause)logiikan lauseeseen ϕ voidaan liittää niiden tulkintojen (eli totuusarvojakeluiden) $\mathcal{T}$ joukko, jotka toteuttavat lauseen ϕ:

$$\mathrm{Mod}(\phi) = \{\mathcal{T} : \mathcal{T}(\phi) = \mathbf{1}\} \tag{17}$$

Näitä tulkintoja sanotaan lauseen ϕ *malleiksi (models)*.

Kukin malli voidaan ymmärtää yhdeksi sellaiseksi asiaintilaksi, joka saattaa lauseen ϕ mukaan vallita todellisuudessa. Kuva 56 osoittaa, mitä joukko-opillisia operaatioita eri loogiset operaatiot eli *konnektiivit* vastaavat.

Nyt kuvassa 55 *lause* ψ *on luvallisesti pääteltävissä lauseesta* ϕ täsmälleen silloin kun

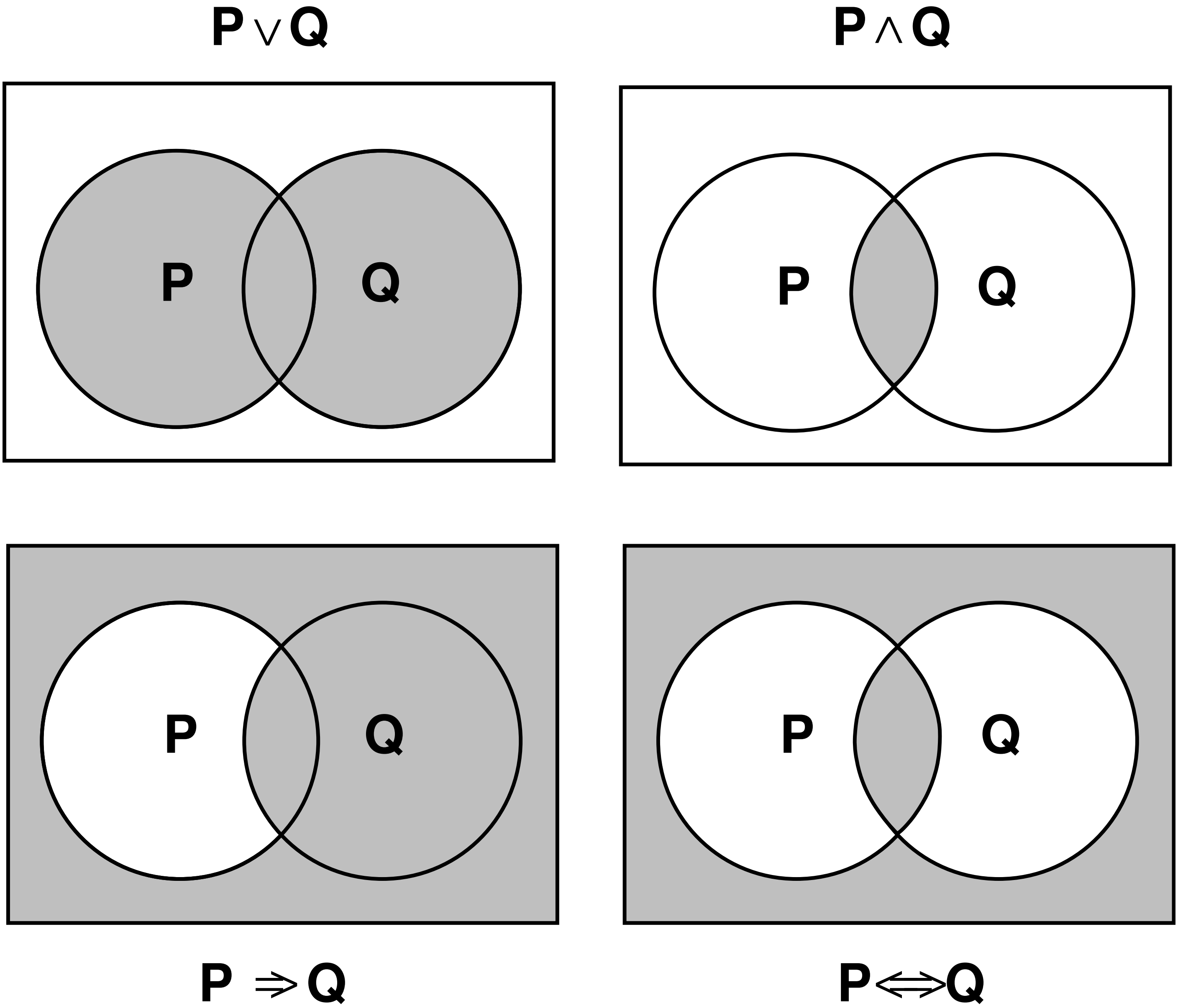
$$\mathrm{Mod}(\phi) \subseteq \mathrm{Mod}(\psi). \tag{18}$$

Nyt ϕ on *vahvempi* väite kuin ψ siitä, miten maailma todellisuudessa on tai ei ole, koska se sulkee pois enemmän vaihtoehtoisia mahdollisia tulkintoja.

Jos siis tiedämme, että ”aurinko paistaa ja vettä sataa”, niin me saamme päätellä siitä sekä ”aurinko paistaa” että ”vettä sataa”.

Implikaatioiden $P \Rightarrow Q$ tärkeys perustuukin seuraavaan järkeilyyn:

$P \Rightarrow Q$ on tautologia	joss	$\mathrm{Mod}(P) \subseteq \mathrm{Mod}(Q)$ [kuvan 56 vasen alakulma]
	joss	takajäsen Q on pääteltävissä etujäsenestä P [yhtälö (18)]
	joss	kun todistusaskel vie lauseesta P lauseeseen Q, niin totuus ei vähene [yhtälöstä (16)].



Kuva 56: Lauselogiikka ja joukko-oppi [RN95, kuva 6.12].

Tietämyskannan KB (ja yleisemmin minkä tahansa lausejoukon eli *teorian (theory)*) malleiksi otetaan ne tulkinnat, jotka toteuttavat samalla kertaa *kaikki* sen sisältämät lauseet eli

$$\begin{aligned}\mathrm{Mod}(KB) &= \bigcap_{\phi \in KB} \mathrm{Mod}(\phi) \\ &= \mathrm{Mod}\left(\bigwedge_{\phi \in KB} \phi\right)\end{aligned}$$

missä jälkimmäinen askel on kuvan 56 oikea yläkulma (joskin se on syntaktisesti oikein vain äärellisille KB).

Toisin sanoen KB on massiivinen *ja*-lause kaikista operaatiolla $\textsc{Tell}$ siihen syötetyistä lauseista. Kukin näistä lauseista rajaa pois joitakin maailman tiloja mahdottomina.

Operaatio $\textsc{Ask}$ taas selvittää päteekö annettu kysymyslause ψ *kaikissa* niissä tiloissa, jotka ovat vielä pysyneet mahdollisina. Se voitaisiin *periaatteessa* toteuttaa selvittämällä, onko

$$\left(\bigwedge_{\phi \in KB} \phi \right) \Rightarrow \psi$$

tautologia. Se taas onnistuisi laskemalla vastaava totuustaulu, ja tarkistamalla, että sen jokainen rivi päätyisi arvoon $\mathbf{1}$.

Tämä olisi kuitenkin eksponentiaalista tietämyskannan KB sisältämien lausemuuttujien lukumäärän m suhteen, koska jo itse taulu sisältäisi 2^m riviä, jotka kaikki pitäisi tarkistaa.

Sukulaiskysymys "onko annettu lauselogiikan kaava χ toteutuva" — eli tuleeko sen totuustauluun yhtään kertaa arvoa $\mathbf{1}$, eli ns. SAT-ongelma — olikin se alkuperäinen ongelma, josta Stephen Cook kehitti vuonna 1971 NP-täydellisyyden käsitteen (luku 1.2) ongelmien laskennallisen vaikeuden luonnehtimiseen. (Artikkelin otsikkokin oli "The Complexity of Theorem Proving Procedures".)

Intuitiivisesti lauselogiikan toteutuvuusongelma kuuluu sellaisiin ongelmiin, jotka ovat helpoimpia niistä ongelmista, jotka ovat luultavasti tietokoneille vaikeita.

Tätä vaikeutta yritetään välttää – sekä ihmisen että koneen takia – kehittämällä joukko oikeellisia päättelysääntöjä ja *etsimällä halutun tuloksen todistusta annetuista lähtötiedoista*, esimerkiksi seuraavasti:

- Etsinnän tila $\mathcal{E}$ on KB ja siitä tähän mennessä päätellyt lauseet.

- Tilaa päivitetään valitsemalla siitä lauseet $\alpha_1, \ldots, \alpha_n \in \mathcal{E}$ jotka sopivat johonkin päättelysääntöön

$$\frac{\alpha_1, \ldots, \alpha_n}{\beta} \tag{19}$$

ja lisäämällä säännön antama tuloslause β tilaan $\mathcal{E}$.

sääntö	nimi
$\dfrac{\alpha \Rightarrow \beta, \qquad \alpha}{\beta}$	modus ponens
$\dfrac{\alpha_1 \wedge \ldots \wedge \alpha_n}{\alpha_i}$	konjunktion poisto
$\dfrac{\alpha_1, \ldots, \alpha_n}{\alpha_1 \wedge \ldots \wedge \alpha_n}$	konjunktion tuonti
$\dfrac{\alpha_i}{\alpha_1 \vee \ldots \vee \alpha_n}$	disjunktion tuonti
$\dfrac{\neg\neg\alpha}{\alpha}$	kaksoisnegaation poisto
$\dfrac{\alpha \vee \beta, \qquad \neg\beta \vee \gamma}{\alpha \vee \gamma}$	resoluutio

Taulukko 7: Lauselogiikan päättelysääntöjä.

- Odotellaan ilmestyykö haluttu seurauslause ψ tilaan $\mathcal{E}$.
- Variaatioita:
  - Jos etsintää suoritetaankin taaksepäin (vrt. luku 3.1.6), niin säännöt (19) luetaankin alhaalta ylös ohjeina miten johtopäätös β voidaan purkaa oletuksiinsa $\alpha_1, \ldots, \alpha_n$.
  - Aloitetaankin tilasta $\mathcal{E} = KB \cup \{\neg\psi\}$ ja odotellaan ilmestyykö $\mathbf{0}$. Jos ilmestyy, niin ei ole yhtään sellaista mallia, jossa koko tietämyskanta KB olisi totta mutta seurauslause ψ ei.
    Tämä on *todistamista ristiriidan kautta (proof by contradiction):* tuloksen kieltäminen on ristiriidassa oletusten kanssa.

Eräs käytännössä tärkeä täyden lauselogiikan osajoukko koostuu