LAUSELOGIIKKA

1. Lauselogiikan kieli
2. Lauselogiikan semantiikka
3. Semanttiset peruskäsitteet
4. Semanttinen taulu
5. Vaihtoehtoisia todistusjärjestelmiä
6. Normaalimuodot
7. Resoluutio
8. Laskennallisesta vaativuudesta

1 Lauselogiikan kieli

- Lauselogiikan aakkosto
- Kielen määritelmä
- Lauseiden muodostamisesta
- Sopimukset sulkujen käytöstä
- Esimerkki: rakenteinen induktio

1.1 Lauselogiikan aakkosto

- atomiset lauseet: $A, A_1, A_2, \ldots, B, B_1, \ldots, C, \ldots$
- negaatiosymboli: $\neg$ (ei)
- konjunktiosymboli: $\wedge$ (ja)
- disjunktiosymboli: $\vee$ (tai)
- implikaatiosymboli: $\rightarrow$ (jos ... niin)
- ekvivalenssisymboli: $\leftrightarrow$ (jos ja vain jos)
- sulut: ()

Symboleja $\neg, \vee, \wedge, \rightarrow$ ja $\leftrightarrow$ kutsutaan *konnektiiveiksi*, koska niiden avulla kytketään yksinkertaisempia lausekkeita (lauseita) monimutkaisemmiksi.

1.2 Kielen määritelmä

Olkoon $\mathcal{P}$ ei-tyhjä joukko atomisia lauseita.

Lauselogiikan lauseenmuodostussäännöt:

1. Jokainen atominen lause $A \in \mathcal{P}$ on *lause*.
2. Jos α ja β ovat lauseita, niin myös $(\neg\alpha)$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \rightarrow \beta)$, $(\alpha \leftrightarrow \beta)$ ovat *lauseita*.
3. vain edellä olevien sääntöjen perusteella muodostetut merkkijonot ovat *lauseita*.

Näiden sääntöjen nojalla muodostettavissa olevien lauseiden joukkoa kutsutaan (atomisten lauseiden joukkoon $\mathcal{P}$ perustuvaksi) lauselogiikan kieleksi $\mathcal{L}$.

Vaihtoehtoinen määritelmä

Olkoon $\mathcal{P}$ ei-tyhjä joukko atomisia lauseita.

Määritelmä. Atomisten lauseiden joukkoon $\mathcal{P}$ perustuva *lauselogiikan kieli* $\mathcal{L}$ on merkkijonojen joukon $(\mathcal{P} \cup \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\})^*$ *pienin* osajoukko, joka on suljettu seuraavien vaatimusten suhteen:

1. Jos $A \in \mathcal{P}$, niin $A \in \mathcal{L}$.
2. Jos $\alpha \in \mathcal{L}$ ja $\beta \in \mathcal{L}$, niin $(\neg\alpha) \in \mathcal{L}$, $(\alpha \vee \beta) \in \mathcal{L}$, $(\alpha \wedge \beta) \in \mathcal{L}$, $(\alpha \rightarrow \beta) \in \mathcal{L}$ ja $(\alpha \leftrightarrow \beta) \in \mathcal{L}$.

Esimerkki. Jos $\mathcal{P} = \{A, B\}$, niin esimerkiksi $A, B, (\neg A), ((\neg A) \vee B)$ ja $(((\neg A) \vee B) \rightarrow A)$ ovat kielen $\mathcal{L}$ lauseita. Sen sijaan merkkijonot $(\neg())$ ja $(A \vee C)$ eivät ole $\mathcal{L}$:n lauseita.

Jokaisella lauselogiikan lauseella on yksikäsitteinen *jäsennyspuu*.

Määritelmä. Määritellään jäsennyspuut lauserakenteen mukaisesti:

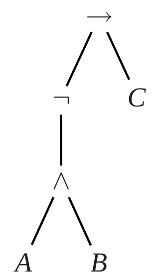
1. Atomisen lauseen $A \in \mathcal{P}$ jäsennyspuu $\mathrm{JP}(A)$ on alla vasemmalla.
2. Negaation $(\neg\alpha)$ jäsennyspuu $\mathrm{JP}(\neg\alpha)$ on annettu keskellä.
3. Jos $*$ on jokin lauselogiikan binäärikonnektiiveista, lauseen $(\alpha * \beta)$ jäsennyspuu $\mathrm{JP}(\alpha * \beta)$ on annettu oikealla.

Yllä $\mathrm{JP}(\alpha)$ ja $\mathrm{JP}(\beta)$ ovat lauseiden α ja β rekursiivisesti määräytyvät jäsennyspuut, jotka sijoitetaan kyseisten jäsennyspuiden *alipuiksi*.

Esimerkki. Lauseen $((\neg(A \wedge B)) \rightarrow C)$ jäsennyspuu on seuraava:

Jäsennyspuun juuressa oleva konnektiivi $\rightarrow$ määrää, että annettu lause on *muodoltaan* implikaatio (tai *implikaatio* lyhyesti sanottuna).

Vastaavasti määritellään lauseet, jotka ovat *negaatioita*, *konjunktioita*, *disjunktioita* ja *ekvivalensseja*.



Lauseiden ominaisuuksia voidaan todistaa induktiolla lauserakenteen suhteen (tai lauseita vastaavien jäsennyspuiden rakenteen suhteen).

1.3 Lauseiden muodostaminen

Jos lähtökohtana on joukko luonnollisen kielen lauseita,

- tunnistetaan atomiset lauseet eli väittämät, joita ei voida enää loogisessa mielessä pilkkoa osiin ja
- tunnistetaan konnektiivit ja muodostetaan vastaavat logiikan lauseet.

Tavoitteena voi olla myös jonkin järjestelmän määrittely suoraan logiikan lausein. Tällöin

- valitaan sopiva joukko järjestelmän ominaisuuksia kuvaavia atomisia lauseita ja
- määritellään näiden väliset suhteet/riippuvuudet logiikan lausein.

Esimerkki. Muotoillaan seuraava luonnollisen kielen lause lauselogiikan lauseena.

Jos tiedosto on liian suuri, niin se tiivistetään tai poistetaan.

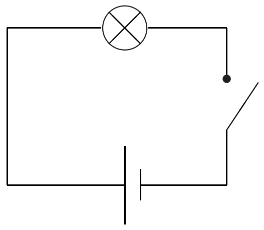
Valitaan atomiset lauseet

$$
\begin{aligned}
A &= \text{“Tiedosto on liian suuri”},\\
B &= \text{“Tiedosto tiivistetään” ja}\\
C &= \text{“Tiedosto poistetaan”}.
\end{aligned}
$$

Saadaan: jos A, niin B tai C.

Tunnistetaan konnektiivit: $(A \rightarrow (B \vee C))$.

Esimerkki. Kuvataan seuraavaa yksinkertaista järjestelmää lauselogiikan lausein.



Valitaan atomisiksi lauseiksi

$$
\begin{aligned}
L &= \text{“Lamppu palaa”},\\
K &= \text{“Kytkin on suljettu” ja}\\
P &= \text{“Paristossa on riittävästi varausta”}.
\end{aligned}
$$

Määritellään järjestelmän sallitut tilat lauseilla

$$((\neg P) \rightarrow (\neg L)) \text{ ja}$$
$$(P \rightarrow (L \leftrightarrow K)).$$

Tulkitse nämä kaksi lausetta luonnolliselle kielelle!

1.4 Sopimukset sulkeiden käytöstä

- Uloimmat sulkeet tapana jättää pois: $A \rightarrow B$ eikä $(A \rightarrow B)$.
- Konnektiivien *presedenssi* eli *sidontajärjestys*:
  1. $\neg$ on vahvin konnektiiveista.
  2. $\vee$ ja $\wedge$ ovat heikompia kuin $\neg$, mutta vahvempia kuin $\rightarrow$ ja $\leftrightarrow$.
  3. $\rightarrow$ ja $\leftrightarrow$ ovat heikoimmat konnektiivit.

  Esimerkiksi: $\neg A \rightarrow B$ eikä $(\neg A) \rightarrow B$,
  $A \wedge B \rightarrow B \vee C$ eikä $(A \wedge B) \rightarrow (B \vee C)$,
  mutta $(A \rightarrow B) \vee (B \leftrightarrow C)$.
- Poikkeuksena ketjudisjunktiot/konjunktiot: kirjoitetaan $A \vee B \vee C$ lauseiden $A \vee (B \vee C)$ ja $(A \vee B) \vee C$ sijaan.

Lisähuomioita sulkeiden käytöstä

- Edellä tehdyt sopimukset (ketjukonjuktioita ja -disjunktioita lukuunottamatta) takaavat, että lauseen ϕ jäsennyspuu säilyy yksikäsitteisenä sulkeita vähennettäessä.
- Esimerkiksi merkkijonoa $A \rightarrow B \rightarrow C$ ei pystytä jäsentämään lauseeksi (yksikäsitteisesti).

  Tarvitaan sulkeet: $A \rightarrow (B \rightarrow C)$ tai $(A \rightarrow B) \rightarrow C$.

  Näillä lauseilla on yksikäsitteiset jäsennyspuut.
- Ketjudisjunktiokin $A \vee B \vee C$ voidaan jäsentää kahdella tavalla: $(A \vee B) \vee C$ tai $A \vee (B \vee C)$.

  Jatkossa näille annetaan kuitenkin sama merkitys, joten on samantekevää kuinka jäsennys suoritetaan.

1.5 Esimerkki: rakenteinen induktio

Määritelmä. Lauseen *alilauseet* määräytyvät seuraavasti:

Atomisen lauseen A ainoa alilause on A.

Lauseen $(\neg\alpha)$ alilauseita ovat α:n alilauseet ja $(\neg\alpha)$.

Lauseen $(\alpha\wedge\beta)$ alilauseita ovat α:n ja β:n alilauseet ja $(\alpha\wedge\beta)$.

Lauseiden $(\alpha\vee\beta)$, $(\alpha\to\beta)$ ja $(\alpha\leftrightarrow\beta)$ alilauseet määritellään samaan tapaan kuin $(\alpha\wedge\beta)$:n.

Esimerkki. Lauseen $A\to B\vee C$ alilauseet ovat $A,B,C,B\vee C$ ja $A\to B\vee C$.

Väite. Lauseen ϕ alilauseita on korkeintaan niin monta kuin on ϕ:ssä esiintyvien atomisten lauseiden lukumäärän ja ϕ:n konnektiivien lukumäärän summa.

Todistetaan väite induktiolla lauserakenteen suhteen.

Otetaan lauseelle ϕ käyttöön seuraavat merkinnät:

- $\#\phi$: lauseen ϕ alilauseiden lukumäärä,
- $A\phi$: lauseessa ϕ esiintyvien atomisten lauseiden lukumäärä ja
- $K\phi$: lauseen ϕ konnektiivien lukumäärä.

Näillä merkinnöillä väite saadaan muotoon $\#\phi\leq A\phi+K\phi$.

Esimerkki. Väittämä pitää paikkansa ainakin lauseen $\phi=(B\to C)\to(B\to C)$ tapauksessa: $\#\phi=4$, $A\phi=2$ ja $K\phi=3$.

Todistus.

Perustapaus: ϕ on atominen lause A.

Koska $\#\phi=1$ määritelmän mukaan, $A\phi=1$ ja $K\phi=0$, väittämä pitää tässä tapauksessa paikkansa.

Induktioaskel: ϕ on muotoa $(\neg\alpha)$.

Määritelmän mukaisesti: $\#(\neg\alpha)=1+\#\alpha$.

Induktio-oletuksen perusteella: $\#\alpha\leq A\alpha+K\alpha$.

Täten
$$\begin{aligned}\#(\neg\alpha) &\leq 1+A\alpha+K\alpha\\ &= A\alpha+(1+K\alpha)\\ &= A(\neg\alpha)+(1+K\alpha)\\ &= A(\neg\alpha)+K(\neg\alpha).\end{aligned}$$

Näin väittämä tuli todistetuksi muotoa $(\neg\alpha)$ oleville lauseille.

Induktioaskel jatkuu: ϕ on muotoa $(\alpha*\beta)$, missä $*$ on mikä tahansa binäärisistä konnektiiveista $\vee$, $\wedge$, $\to$ ja $\leftrightarrow$.

$$\begin{aligned}\#(\alpha*\beta) &= 1+\#\alpha+\#\beta-\#(\alpha,\beta)\\ &\leq 1+A\alpha+K\alpha+A\beta+K\beta-\#(\alpha,\beta) \quad \text{(ind.-oletus)}\\ &= A\alpha+A\beta-\#(\alpha,\beta)+1+K\alpha+K\beta\\ &\leq A\alpha+A\beta-A(\alpha,\beta)+K(\alpha*\beta)\\ &= A(\alpha*\beta)+K(\alpha*\beta).\end{aligned}$$

Merkintä $\#(\alpha,\beta)$ tarkoittaa lauseiden α ja β yhteisten alilauseiden lukumäärää ja $A(\alpha,\beta)$ lauseiden α ja β yhteisten atomisten lauseiden lukumäärää. Tällöin pätee $A(\alpha,\beta)\leq\#(\alpha,\beta)$.

Näin ollen väittämä tuli todistetuksi kaikille lauseille ϕ.

2 Lauselogiikan semantiikka

- Perustotuustaulukot
- Konnektiivien keskinäisestä määriteltävyydestä
- Lauselogiikan totuusmääritelmä
- Vaihtoehtoinen totuusmääritelmä: valuaatiot

2.1 Perustotuustaulukot

Perustotuustaulukoilla määritellään eri muotoa olevien lauseiden totuusarvot alilauseidensa funktiona.

α	$(\neg\alpha)$
T	E
E	T

α	β	$(\alpha \wedge \beta)$
T	T	T
T	E	E
E	T	E
E	E	E

α	β	$(\alpha \vee \beta)$
T	T	T
T	E	T
E	T	T
E	E	E

Totuustaulukot on helppo sisäistää muistisääntöjen avulla: esim. $(\alpha \wedge \beta)$ on tosi $\iff$ α on tosi **ja** β on tosi.

Huomio. Disjunktio $(\alpha \vee \beta)$ on tosi $\iff$ α on tosi **tai** β on tosi. Disjunktio $(\alpha \vee \beta)$ on siis tosi myös silloin, kun sekä α että β ovat tosia.

Voidaan määritellä myös poissulkeva disjuktio $(\alpha \underline{\vee} \beta)$, joka on epätosi molempien disjunktien ollessa tosia.

α	β	$(\alpha \underline{\vee} \beta)$
T	T	E
T	E	T
E	T	T
E	E	E

$(\alpha \underline{\vee} \beta)$ on tosi $\iff$ **joko** α on tosi **tai** β on tosi.

Perustotuustaulukot (jatkoa)

α	β	$(\alpha \rightarrow \beta)$
T	T	T
T	E	E
E	T	T
E	E	T

α	β	$(\alpha \leftrightarrow \beta)$
T	T	T
T	E	E
E	T	E
E	E	T

$(\alpha \rightarrow \beta)$ on tosi $\iff$ α on epätosi **tai** β on tosi
$\iff$ **jos** α on tosi, **niin** β on tosi.

Huomio. Implikaatio $\rightarrow$ *ei* edellytä syy-seuraus-suhdetta. Esim.

$A =$ "Ruotsi sijaitsee Aasiassa", $B =$ "Joulupukki on olemassa".

Lause $(A \rightarrow B)$ on tosi, koska A on epätosi.

Perustotuustaulukoiden avulla voidaan muodostaa totuustaulukko mille tahansa lauseelle (tai jopa lausejoukolle).

Esimerkki. Tutkittava lause: $(\neg B \wedge (A \rightarrow B))$
Alilauseet: $A, B, \neg B, (A \rightarrow B), (\neg B \wedge (A \rightarrow B))$
Totuustaulukossa on $2^2 = 4$ riviä ja 5 saraketta.

A	B	$\neg B$	$(A \rightarrow B)$	$(\neg B \wedge (A \rightarrow B))$
T	T	E	T	E
T	E	T	E	E
E	T	E	T	E
E	E	T	T	T

Muistathan, että jokaisessa lauseessa ϕ on korkeintaan niin monta alilausetta kuin lauseessa ϕ on atomisia lauseita ja konnektiiveja!

2.2 Konnektiivien keskinäisestä määriteltävyydestä

Esimerkki. Tarkastellaan lauseille $(\alpha \wedge \beta)$ ja $\neg(\neg\alpha \vee \neg\beta)$ muodostettua yhteistä totuustaulukkoa.

α	β	$\alpha \wedge \beta$	$\neg\alpha$	$\neg\beta$	$\neg\alpha \vee \neg\beta$	$\neg(\neg\alpha \vee \neg\beta)$
T	T	T	E	E	E	T
T	E	E	E	T	T	E
E	T	E	T	E	T	E
E	E	E	T	T	T	E

Koska lauseiden $\alpha \wedge \beta$ ja $\neg(\neg\alpha \vee \neg\beta)$ sarakkeissa on identtiset totuusarvot, lauseet ovat semantiikan kannalta samaistettavissa.

$\Longrightarrow$ Lause $\alpha \wedge \beta$ voidaan (tarvittaessa) määritellä syntaktisena lyhennysmerkintänä lauseelle $\neg(\neg\alpha \vee \neg\beta)$.

Konnektiivien tyypillisiä määritelmiä

Konnektiiveja voidaan määritellä toistensa avulla esim. seuraavasti:

1. $(\alpha \wedge \beta)$ lauseena $\neg(\neg\alpha \vee \neg\beta)$
2. $(\alpha \vee \beta)$ lauseena $\neg(\neg\alpha \wedge \neg\beta)$
3. $(\alpha \underline{\vee} \beta)$ lauseena $(\alpha \leftrightarrow \neg\beta)$ tai lauseena $\neg(\alpha \leftrightarrow \beta)$
4. $(\alpha \rightarrow \beta)$ lauseena $(\neg\alpha \vee \beta)$ tai lauseena $\neg(\alpha \wedge \neg\beta)$
5. $(\alpha \leftrightarrow \beta)$ lauseena $((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$ tai lauseena $((\neg\alpha \vee \beta) \wedge (\alpha \vee \neg\beta))$

$\Longrightarrow$ Kaikkia peruskonnektiiveista $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ ei välttämättä tarvita.

Konnektiivien riittävyys

Tarvittaessa voidaan rajoittua seuraaviin konnektiiveihin:

- $\neg$ ja $\vee$ riittävät muiden konnektiivien ($\wedge$, $\rightarrow$, $\leftrightarrow$, $\underline{\vee}$) määrittelemiseen,
- $\neg$ ja $\wedge$ riittävät myös,
- $\neg$ ja $\rightarrow$ riittävät myös ja
- $\perp$ (aina epätosi lause) ja $\rightarrow$ riittävät myös.

Huomio. Toistaiseksi käsitellyt konnektiivit eivät ole ainoat mahdolliset. Esimerkiksi *binäärikonnektiiveja* $*$, jotka kytkevät kaksi lausetta α ja β lauseeksi $\alpha * \beta$, voidaan olennaisesti määritellä $2^{(2\times 2)} = 16$ erilaista.

Yksittäisten konnektiivien riittävyys/riittämättömyys

Huomio. On myös konnektiiveja, jotka riittävät yksinään muiden määrittelemiseen:

- Peircen nuoli: $(\alpha \downarrow \beta) \equiv \neg(\alpha \vee \beta)$
- Shefferin viiva: $(\alpha \mid \beta) \equiv \neg(\alpha \wedge \beta)$

Esimerkki. Implikaatio ei riitä yksinään muiden konnektiivien kuten esimerkiksi negaation määrittelemiseen.

Tarkastellaan lauseita ϕ, jotka on muodostettu käyttäen ainoastaan konnektiivia $\rightarrow$, atomista lausetta A ja sulkeita aakkosina.

Voidaan osoittaa rakenteisella induktiolla, että lauseen ϕ totuustaulun sarake on identtinen joko lauseen A tai lauseen $A \vee \neg A$ kanssa.

Näinollen mikään lauseista ϕ ei voi ilmaista lausetta $\neg A$.

2.3 Lauselogiikan totuusmääritelmä

Määritelmä. *Totuusjakelu* $\mathcal{A}$ on atomisten lauseiden joukon $\mathcal{P}$ osajoukko.

Ajatuksena on, että

- $\mathcal{A}$:n atomiset lauseet ovat tosia,
- $\mathcal{P} - \mathcal{A}$:n atomiset lauseet ovat epätosia.

Huomioita.

- Jos $\mathcal{P}$ on äärellinen, erilaisia totuusjakeluja on $2^{|\mathcal{P}|}$ kappaletta.
- Jokainen totuusjakelu vastaa yhtä totuustaulukon riviä (ja kääntäen).

Totuusjakelu voidaan ymmärtää yhden *asiaintilan* kuvauksena.

Esimerkki. (vrt. aikaisempi lamppuesimerkki)

$\mathcal{A}_1 = \{P\}$: Patterissa on riittävästi varausta.
Kytkin ei ole suljettu.
Lamppu ei pala.

$\mathcal{A}_2 = \{L, K\}$: Patterissa ei ole riittävästi varausta.
Lamppu palaa.
Kytkin on suljettu.

Näistä jälkimmäinen on mitä ilmeisimmin fyysisesti mahdoton, mutta kuitenkin loogisesti mahdollinen asiaintila.

Määritelmä. Seuraavassa määritellään milloin mielivaltainen lause $\phi \in \mathcal{L}$ on *tosi* totuusjakelussa $\mathcal{A}$ (merk. $\mathcal{A} \models \phi$) ja milloin ϕ on *epätosi* totuusjakelussa $\mathcal{A}$ (merk. $\mathcal{A} \not\models \phi$).

1. $\mathcal{A} \models A \iff A \in \mathcal{A}$ (atomisille lauseille $A \in \mathcal{P}$).
2. $\mathcal{A} \models \neg\alpha \iff \mathcal{A} \not\models \alpha$.
3. $\mathcal{A} \models \alpha \wedge \beta \iff \mathcal{A} \models \alpha$ ja $\mathcal{A} \models \beta$.
4. $\mathcal{A} \models \alpha \vee \beta \iff \mathcal{A} \models \alpha$ tai $\mathcal{A} \models \beta$.
5. $\mathcal{A} \models \alpha \rightarrow \beta \iff \mathcal{A} \not\models \alpha$ tai $\mathcal{A} \models \beta$.
6. $\mathcal{A} \models \alpha \leftrightarrow \beta \iff$ joko $\mathcal{A} \models \alpha$ ja $\mathcal{A} \models \beta$, tai $\mathcal{A} \not\models \alpha$ ja $\mathcal{A} \not\models \beta$.

Huomio. Esim. kohdan 3 nojalla $\mathcal{A} \not\models \alpha \wedge \beta \iff \mathcal{A} \not\models \alpha$ tai $\mathcal{A} \not\models \beta$.

Esimerkki. Olkoon $\mathcal{A} = \emptyset$ totuusjakelu. Totuusmääritelmän nojalla:

$$\mathcal{A} \not\models A,\ \mathcal{A} \not\models B,\ \mathcal{A} \models \neg B,\ \mathcal{A} \models A \rightarrow B \text{ ja } \mathcal{A} \models \neg B \wedge (A \rightarrow B).$$

Vertaa tätä laskelmaa kalvon 21 totuustaulukon viimeiseen riviin.

Totuusmääritelmän seurauksia

Väite. Jos $\mathcal{A} \subseteq \mathcal{P}$ on totuusjakelu, niin kaikille lauseille $\phi \in \mathcal{L}$, joko $\mathcal{A} \models \phi$ tai $\mathcal{A} \not\models \phi$.

Merkitään At(ϕ):llä ϕ:ssä esiintyvien atomisten lauseiden joukkoa.

Väite. Olkoon $\mathcal{A}_1 \subseteq \mathcal{P}$ ja $\mathcal{A}_2 \subseteq \mathcal{P}$ kaksi totuusjakelua ja $\phi \in \mathcal{L}$ lause.

Jos $\mathcal{A}_1 \cap \mathrm{At}(\phi) = \mathcal{A}_2 \cap \mathrm{At}(\phi)$, niin $\mathcal{A}_1 \models \phi \iff \mathcal{A}_2 \models \phi$.

Nämä voidaan todistaa induktiolla ϕ:n rakenteen suhteen.

2.4 Vaihtoehtoinen totuusmääritelmä: valuaatiot

(Tätä määritelmää käytetään Neroden ja Shoren kirjassa).

Määritelmä. *Valuaatio* $\mathcal{V}$ on *konnektiivien totuustaulukkoja noudattava* funktio kielen $\mathcal{L}$ *lauseiden joukolta* joukolle $\{T, E\}$.

Valuaatioilla ja totuusjakeluilla on seuraava yhteys:

1. Olkoon $\mathcal{V}$ valuaatio ja $\mathcal{A} = \{A \in \mathcal{P} \mid \mathcal{V}(A) = T\}$.
   Tällöin kaikille lauseille $\phi \in \mathcal{L}$ pätee:
   $\mathcal{A} \models \phi \iff \mathcal{V}(\phi) = T$.
2. Jos $\mathcal{A}$ on totuusjakelu, niin on olemassa yksikäsitteinen valuaatio $\mathcal{V}$ siten että $\mathcal{A} = \{A \in \mathcal{P} \mid \mathcal{V}(A) = T\}$.

3 Semanttiset peruskäsitteet

- Mallin käsite
- Toteutuvuus
- Pätevyys
- Looginen seuraavuus
- Looginen ekvivalenssi
- Peruskäsitteiden väliset yhteydet
- Loogisten seurauksien ominaisuuksia
- Tietämyksen esittämisestä

3.1 Mallin käsite

Määritelmä. Totuusjakelu $\mathcal{A} \subseteq \mathcal{P}$ on lauseen $\alpha \in \mathcal{L}$ *malli*, joss lause α on tosi $\mathcal{A}$:ssa eli $\mathcal{A} \models \alpha$.

Esimerkki. Totuusjakelut $\mathcal{A}_1 = \{A\}, \mathcal{A}_2 = \{B\}$ ja $\mathcal{A}_3 = \{A, B\}$ ovat malleja lauseelle $A \vee B$.

Määritelmä. Totuusjakelu $\mathcal{A} \subseteq \mathcal{P}$ on lausejoukon $\Sigma \subseteq \mathcal{L}$ *malli* (merk. $\mathcal{A} \models \Sigma$), joss kaikille lausejoukon Σ lauseille $\sigma \in \Sigma$ pätee $\mathcal{A} \models \sigma$.

Näin ollen lausejoukon Σ lauseet tulkitaan konjunktiivisesti.

Esimerkki. Olkoon $\mathcal{P} = \{P, L, K\}$. Laaditaan lamppuesimerkin lausejoukolle $\Sigma = \{\neg P \rightarrow \neg L, P \rightarrow (L \leftrightarrow K)\} \subseteq \mathcal{L}$ totuustaulukko:

P	L	K	$\neg P$	$\neg L$	$\neg P \rightarrow \neg L$	$L \leftrightarrow K$	$P \rightarrow (L \leftrightarrow K)$	
T	T	T	E	E	$\boxed{T}$	T	$\boxed{T}$	$\longleftarrow$
T	T	E	E	E	T	E	E	
T	E	T	E	T	T	E	E	
T	E	E	E	T	$\boxed{T}$	T	$\boxed{T}$	$\longleftarrow$
E	T	T	T	E	E	T	T	
E	T	E	T	E	E	E	T	
E	E	T	T	T	$\boxed{T}$	E	$\boxed{T}$	$\longleftarrow$
E	E	E	T	T	$\boxed{T}$	T	$\boxed{T}$	$\longleftarrow$

Malleja on siis neljä erilaista vastaten järjestelmän sallittuja tiloja.

3.2 Toteutuvuus

Määritelmä. Lause $\alpha \in \mathcal{L}$ (tai lausejoukko $\Sigma \subseteq \mathcal{L}$) on *toteutuva*, joss ainakin yksi totuusjakelu $\mathcal{A} \subseteq \mathcal{P}$ on sen malli.

Huomio. Koska lauseiden totuusarvot määräytyvät niissä esiintyvien atomisten lauseiden totuusarvoista, voimme rajoittua totuusjakeluihin $\mathcal{A} \subseteq \mathrm{At}(\phi)$, missä $\mathrm{At}(\phi)$ on lauseessa ϕ esiintyvien atomisten lauseiden joukko, tai totuusjakeluihin $\mathcal{A} \subseteq \mathrm{At}(\Sigma) = \cup\{\mathrm{At}(\sigma) | \sigma \in \Sigma\}$ selvittäessämme lauseen ϕ tai lausejoukon Σ malleja.

- *Toteutumattomalla* lauseella/lausejoukolla ei ole yhtään mallia.
- Lauseen toteutuvuuden selvittäminen on yksi keskeisimmistä logiikkaan liittyvistä laskennallisista ongelmista.

Toteutuvuuden tutkiminen totuustaulukolla:

- muodostetaan α:lle (Σ:lle) totuustaulukko ja
- tarkastetaan onko α tosi (kaikki Σ:n lauseet tosia) jollakin rivillä.

Esimerkki.

Onko $A \wedge \neg A$ toteutuva?

A	$\neg A$	$A \wedge \neg A$
T	E	E
E	T	E

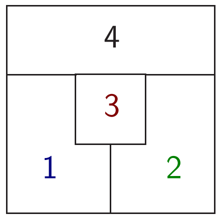
Ei.

Onko $A \vee \neg B$ toteutuva?

A	B	$\neg B$	$A \vee \neg B$
T	T	E	$\boxed{T}$
T	E	T	$\boxed{T}$
E	T	E	E
E	E	T	$\boxed{T}$

Kyllä.

Esimerkki. (Rajoiteohjelmointi) Tarkastellaan oheisen kuvan mukaisen kartan värittämistä kolmella eri värillä siten, että vierekkäisillä alueilla on eri värit. Oheiset lauseet kuvaavat kolmiväritystä vastaavat rajoitteet.



Jokaiselle alueelle i värin valitsevat lauseet:

$P_i \vee V_i \vee S_i$,

$\neg(P_i \wedge V_i)$, $\neg(V_i \wedge S_i)$ ja $\neg(S_i \wedge P_i)$.

Jokaiselle alueparille $\langle i, j\rangle$ $(i < j)$ rajoitteet:

$\neg(P_i \wedge P_j)$, $\neg(V_i \wedge V_j)$ ja $\neg(S_i \wedge S_j)$.

Totuusarvojakeluja (mallikandidaatteja) on yhteensä $2^{12} = 4096$ kpl!

Toisaalta alueet voidaan värittää $3^4 = 81$ eri tavalla.

Tämä lausejoukko on toteutumaton $\Longrightarrow$ *kolmiväritys on mahdoton*.

Muistanet, että neljä väriä riittää aina tasograafin värittämiseen.

3.3 Pätevyys

Määritelmä. Lause $\alpha \in \mathcal{L}$ on *pätevä/tautologia* (merkitään $\models \alpha$), jos $\mathcal{A} \models \alpha$ kaikille totuusjakeluille $\mathcal{A} \subseteq \mathcal{P}$.

Esimerkki. Olkoon $\mathcal{P} = \{A\}$ ja $\mathcal{L}$ vastaava kieli. Lause $A \vee \neg A$ on pätevä, koska $A \vee \neg A$ on tosi totuusjakeluissa $\mathcal{A}_1 = \{\}$ ja $\mathcal{A}_2 = \{A\}$.

Vastamallit

Tarkastellaan mielivaltaista lausetta $\phi \in \mathcal{L}$.

- Jos ϕ on pätevä, $\mathcal{A} \models \phi$ kaikille totuusjakeluille $\mathcal{A} \subseteq \mathcal{P}$.
- Jos ϕ ei ole pätevä, löytyy totuusjakelu $\mathcal{A}$ siten, että $\mathcal{A} \not\models \phi$.

Jälkimmäisessä tapauksessa kutsumme totuusjakelua $\mathcal{A}$ *vastamalliksi* (tai vastaesimerkiksi) lauseen ϕ pätevyydelle.

Pätevyyden tutkiminen totuustaulukolla:

- muodostetaan α:lle totuustaulukko ja
- tarkistetaan, että α on tosi jokaisella rivillä.

Esimerkki.

Onko $A \wedge B \rightarrow A$ pätevä?

A	B	$A \wedge B$	$A \wedge B \rightarrow A$
T	T	T	T
T	E	E	T
E	T	E	T
E	E	E	T

Kyllä.

Onko $A \vee B \rightarrow A$ pätevä?

A	B	$A \vee B$	$A \vee B \rightarrow A$
T	T	T	T
T	E	T	T
E	T	T	$\boxed{E}$
E	E	E	T

Ei.

3.4 Looginen seuraavuus

Määritelmä. Lause $\alpha \in \mathcal{L}$ on lausejoukon $\Sigma \subseteq \mathcal{L}$ *looginen seuraus* (merkitään $\Sigma \models \alpha$), jos $\mathcal{A} \models \alpha$ kaikille lausejoukon Σ malleille $\mathcal{A} \subseteq \mathcal{P}$.

Esimerkki. $\neg\neg A$ seuraa loogisesti lausejoukoista $\{A\}$ ja $\{A \wedge \neg A\}$.

1. Jos $\mathcal{A} \models \{A\}$, niin välttämättä $\mathcal{A} \models A$, jolloin myös $\mathcal{A} \models \neg\neg A$.
2. Ei löydy totuusjakelua $\mathcal{A}$ siten, että $\mathcal{A} \models \{A \wedge \neg A\}$.

Käytämme *vastamallin* käsitettä myös loogisen seuraavuuden yhteydessä.

- Jos $\Sigma \not\models \phi$, lausejoukolla Σ on malli $\mathcal{A}$ siten, mutta $\mathcal{A} \not\models \phi$.

Esimerkki. $\{A, B \rightarrow A\} \not\models \neg\neg B$, koska löytyy vastamalli $\mathcal{A} = \{A\}$ siten, että $\mathcal{A} \models \{A, B \rightarrow A\}$ ja $\mathcal{A} \not\models \neg\neg B$.

Loogisen seuraavuuden tutkiminen totuustaulukolla:

- muodostetaan lausejoukolle $\Sigma \cup \{\alpha\}$ totuustaulukko,
- todetaan rivit, joilla kaikki Σ:n lauseet ovat tosia (Σ:n mallit) ja
- tarkistetaan, että α on näillä riveillä tosi.

Esimerkki. Todetaan $\{A, (A \rightarrow D)\} \models D$ ja $\{(A \rightarrow B)\} \not\models B$ vastaavista totuustaulukoista:

A	D	$(A \rightarrow D)$	
T	T	T	$\longleftarrow$
T	E	E	
E	T	T	
E	E	T	

A	B	$(A \rightarrow B)$	
T	T	T	$\longleftarrow$
T	E	E	
E	T	T	$\longleftarrow$
E	$\boxed{E}$	T	$\longleftarrow$

Huomio. Kaikille lausejoukoille Σ pätee $\Sigma \models A \vee \neg A$.

Esimerkki. Tutkitaan, onko lause $\neg L \vee K$ looginen seuraus lamppuesimerkin lausejoukolle $\Sigma = \{P \rightarrow (L \leftrightarrow K), \neg P \rightarrow \neg L\}$. On.

P	L	K	$\neg P$	$\neg L$	$\neg P \rightarrow \neg L$	$L \leftrightarrow K$	$P \rightarrow (L \leftrightarrow K)$	$\neg L \vee K$	
T	T	T	E	E	T	T	T	$\boxed{T}$	$\longleftarrow$
T	T	E	E	E	T	E	E	E	
T	E	T	E	T	T	E	E	T	
T	E	E	E	T	T	T	T	$\boxed{T}$	$\longleftarrow$
E	T	T	T	E	E	T	T	T	
E	T	E	T	E	E	E	T	E	
E	E	T	T	T	T	E	T	$\boxed{T}$	$\longleftarrow$
E	E	E	T	T	T	T	T	$\boxed{T}$	$\longleftarrow$

3.5 Looginen ekvivalenssi

Määritelmä. Lauseet $\alpha \in \mathcal{L}$ ja $\beta \in \mathcal{L}$ ovat *loogisesti ekvivalentteja* ($\alpha \equiv \beta$), joss $\mathcal{A} \models \alpha \iff \mathcal{A} \models \beta$ kaikille totuusjakeluille $\mathcal{A} \subseteq \mathcal{P}$.

Esimerkki. A ja $\neg\neg A$ ovat loogisesti ekvivalentit, koska näillä on sama totuusarvo kaikissa totuusjakeluissa.

Väite. Lauseet $\alpha \in \mathcal{L}$ ja $\beta \in \mathcal{L}$ ovat loogisesti ekvivalentit, joss niillä on täsmälleen samat mallit $\mathcal{A} \subseteq \mathcal{P}$.

Lauseiden loogisen ekvivalenssin selvittäminen:

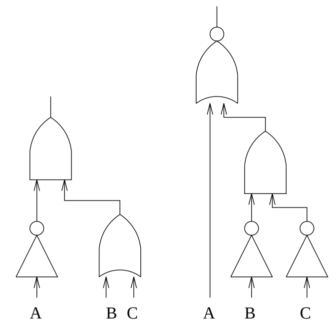
- muodostetaan lauseille α ja β yhteinen totuustaulukko ja
- tarkastetaan, että α:lla ja β:lla on sama totuusarvo jokaisella rivillä.

Esimerkki. Ovatko lauseet $A \rightarrow B$ ja $\neg B \rightarrow \neg A$ loogisesti ekvivalentit?

A	B	$\neg A$	$\neg B$	$A \rightarrow B$	$\neg B \rightarrow \neg A$
T	T	E	E	T	T
T	E	E	T	E	E
E	T	T	E	T	T
E	E	T	T	T	T

Kyllä.

Esimerkki. Palataan johdannon esimerkkiin, jossa ongelmana oli selvittää laskevatko seuraavat kombinatoriset piirit samat funktiot:



Piirit voidaan esittää lauselogiikalla seuraavina lauseina

$$\phi = \neg A \wedge (B \vee C) \text{ ja } \psi = \neg(A \vee (\neg B \wedge \neg C)).$$

Esimerkki. Ratkaistaan ongelma osoittamalla, että piirejä vastaavat lauseet ovat loogisesti ekvivalentit.

A	B	C	$\neg A$	$\neg B$	$\neg C$	$B \vee C$	ϕ	$\neg B \wedge \neg C$	$A \vee (\neg B \wedge \neg C)$	ψ
T	T	T	E	E	E	T	E	E	T	E
T	T	E	E	E	T	T	E	E	T	E
T	E	T	E	T	E	T	E	E	T	E
T	E	E	E	T	T	E	E	T	T	E
E	T	T	T	E	E	T	T	E	E	T
E	T	E	T	E	T	T	T	E	E	T
E	E	T	T	T	E	T	T	E	E	T
E	E	E	T	T	T	E	T	T	T	E

Koska lauseita ϕ ja ψ vastaavat sarakkeet ovat identtiset, ne ovat loogiseseti ekvivalentit. Tästä seuraa, että piirit laskevat samat funktiot.

Määritelmä. Lausejoukot $\Sigma_1 \subseteq \mathcal{L}$ ja $\Sigma_2 \subseteq \mathcal{L}$ ovat *loogisesti ekvivalentit* (merk. $\Sigma_1 \equiv \Sigma_2$), joss kaikille totuusjakeluille $\mathcal{A} \subseteq \mathcal{P}$ pätee seuraavaa:

$$\mathcal{A} \models \sigma_1 \text{ kaikille } \sigma_1 \in \Sigma_1 \iff \mathcal{A} \models \sigma_2 \text{ kaikille } \sigma_2 \in \Sigma_2.$$

Väite. Lausejoukot $\Sigma_1 \subseteq \mathcal{L}$ ja $\Sigma_2 \subseteq \mathcal{L}$ ovat loogisesti ekvivalentit, mikäli niillä on täsmälleen samat mallit $\mathcal{A} \subseteq \mathcal{P}$.

Lausejoukkojen loogisen ekvivalenssin selvittäminen: *joko*

- todetaan, että $\mathcal{A} \models \Sigma_2$ kaikille lausejoukon Σ_1 malleille $\mathcal{A}$ ja
- todetaan, että $\mathcal{A} \models \Sigma_1$ kaikille lausejoukon Σ_2 malleille $\mathcal{A}$,

tai vaihtoehtoisesti

- todetaan, että $\Sigma_1 \models \sigma_2$ kaikille lauseille $\sigma_2 \in \Sigma_2$ ja
- todetaan, että $\Sigma_2 \models \sigma_1$ kaikille lauseille $\sigma_1 \in \Sigma_1$.

Esimerkki. Kirjoitetaan lamppuesimerkin järjestelmälle vaihtoehtoinen määritelmä $\Sigma' = \{L \to P, P \wedge L \to K, P \wedge K \to L\}$.

P	L	K	$L \to P$	$P \wedge L$	$P \wedge L \to K$	$P \wedge K$	$P \wedge K \to L$	
T	T	T	$\boxed{T}$	T	$\boxed{T}$	T	$\boxed{T}$	$\longleftarrow$
T	T	E	T	T	E	E	T	
T	E	T	T	E	T	T	E	
T	E	E	$\boxed{T}$	E	$\boxed{T}$	E	$\boxed{T}$	$\longleftarrow$
E	T	T	E	E	T	E	T	
E	T	E	E	E	T	E	T	
E	E	T	$\boxed{T}$	E	$\boxed{T}$	E	$\boxed{T}$	$\longleftarrow$
E	E	E	$\boxed{T}$	E	$\boxed{T}$	E	$\boxed{T}$	$\longleftarrow$

$\Longrightarrow$ $\Sigma' \equiv \Sigma$, koska mallit ovat samat kuin alkuperäisellä määritelmällä Σ.

3.6 Peruskäsitteiden väliset yhteydet

Olkoon $\mathcal{L}$ atomisten lauseiden joukkoon $\mathcal{P}$ perustuva kieli.

Tarkastellaan jatkossa tämän kielen lauseita.

Looginen ekvivalenssi liittyy läheisesti pätevyyteen:

- $\alpha \equiv \beta \iff \models \alpha \leftrightarrow \beta$.

  Todistus.

  $\alpha \not\equiv \beta$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \models \alpha$ ja $\mathcal{A} \not\models \beta$, tai $\mathcal{A} \not\models \alpha$ ja $\mathcal{A} \models \beta$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \not\models \alpha \leftrightarrow \beta$
  $\iff \not\models \alpha \leftrightarrow \beta$.

Pätevyydellä ja loogisella seuraavuudella on kiinteät yhteydet.

- $\models \alpha \iff \emptyset \models \alpha$.
  Seuraa helposti, koska kaikki totuusjakelut $\mathcal{A} \subseteq \mathcal{P}$ ovat tyhjän lausejoukon $\emptyset$ malleja.
- $\{\phi_1, \ldots, \phi_n\} \models \phi \iff \models \phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi$.
  Todistus.
  $\{\phi_1, \ldots, \phi_n\} \not\models \phi$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A}$ on $\{\phi_1, \ldots, \phi_n\}$:n malli ja $\mathcal{A} \not\models \phi$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\forall i \in \{1, \ldots, n\} : \mathcal{A} \models \phi_i$ ja $\mathcal{A} \not\models \phi$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \models \phi_1 \wedge \cdots \wedge \phi_n$ ja $\mathcal{A} \not\models \phi$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \not\models \phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi$
  $\iff \not\models \phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi$.

Pätevyyden ja loogisen seuraavuuden yhteys toteutuvuuteen:

- $\models \alpha \iff \neg\alpha$ on toteutumaton.
  Todistus. Erikoistapaus seuraavasta.
- $\Sigma \models \alpha \iff \Sigma \cup \{\neg\alpha\}$ on toteutumaton.
  Todistus.
  $\Sigma \not\models \alpha$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \models \Sigma$ ja $\mathcal{A} \not\models \alpha$
  $\iff \exists \mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \models \Sigma \cup \{\neg\alpha\}$
  $\iff \Sigma \cup \{\neg\alpha\}$ on toteutuva
  $\iff \Sigma \cup \{\neg\alpha\}$ ei ole toteutumaton.

Edellä esitetyt yhteydet mahdollistavat lauselogiikan päättelyongelmien väliset muunnokset. Esim. loogisen ekvivalenssin, pätevyyden ja loogisen seuraavuuden tutkiminen voidaan palauttaa toteutuvuuden tutkimiseen.

3.7 Loogisten seurauksien ominaisuuksia

Väite. (Kompaktius). Jos $\Sigma \models \phi$, niin on olemassa äärellinen osajoukko $\Sigma' \subseteq \Sigma$ siten, että $\Sigma' \models \phi$.

Todistus. Sivuutetaan.

Määritelmä. Lausejoukon $\Sigma \subseteq \mathcal{L}$ loogisten seurausten joukko on $\mathrm{Cn}(\Sigma) = \{\phi \in \mathcal{L} \mid \Sigma \models \phi\}$.

Loogisten seurauksien joukolla $\mathrm{Cn}(\Sigma)$ on seuraavat perusominaisuudet:

- $\Sigma \subseteq \mathrm{Cn}(\Sigma)$.
- Monotonisuus: $\Sigma_1 \subseteq \Sigma_2 \implies \mathrm{Cn}(\Sigma_1) \subseteq \mathrm{Cn}(\Sigma_2)$.
- $\mathrm{Cn}(\mathrm{Cn}(\Sigma)) = \mathrm{Cn}(\Sigma)$.
- $\Sigma \equiv \mathrm{Cn}(\Sigma)$.

Lisähuomioita loogisesta seuraavuudesta

- Jos $\Sigma \models \phi$, niin $\mathrm{Cn}(\Sigma) = \mathrm{Cn}(\Sigma \cup \{\phi\})$.
  Jos tavoitteena on lausejoukon koon minimointi, lausejoukon loogisia seurauksia ei siis kannata lisätä lausejoukkoon!
- Oletetaan, että $\Sigma \not\models \phi$ ja että lausejoukko Σ halutaan laajentaa lausejoukoksi Σ' siten, että $\Sigma \subseteq \Sigma'$ ja $\Sigma' \models \phi$.
  Apuna voidaan käyttää vastamallia/vastamalleja $\mathcal{A}$, joille $\mathcal{A} \models \Sigma$ ja $\mathcal{A} \not\models \phi$: lausejoukkoon Σ lisättävän lauseen ψ tulisi sulkea pois vastamalli/vastamalleja $\mathcal{A}$, ts. $\mathcal{A} \not\models \psi$.
  Lause ϕ toteuttaa myös tämän vaatimuksen, mutta lausetta ϕ ei välttämättä haluta liittää eksplisiittiseen tietämykseen.

Esimerkki. Lamppuesimerkissä $\Sigma = \{P \wedge L \rightarrow K, P \wedge K \rightarrow L\}$, lause $\phi = \neg L \vee K$, vastamalli $\mathcal{A} = \{L\}$ ja lisättävä lause $\psi = L \rightarrow P$.

3.8 Tietämyksen esittämisestä

- Jokainen lausejoukko Σ määrittää joukon *malleja*, eli totuusjakeluja $\mathcal{A}$, joissa lausejoukon kaikki lauseet ovat tosia.

  Esimerkki. Lamppuesimerkin lausejoukolla
  $$\Sigma = \{\neg P \rightarrow \neg L, P \rightarrow (L \leftrightarrow K)\} \subseteq \mathcal{L}$$
  on mallit $\mathcal{A}_1 = \{P,L,K\}$, $\mathcal{A}_2 = \{P\}$, $\mathcal{A}_3 = \{K\}$ ja $\mathcal{A}_4 = \{\}$.

- Lausejoukon Σ mallit puolestaan määräävät lausejoukon loogisten seurauksien joukon $\mathrm{Cn}(\Sigma)$.

  Esimerkki. Lamppuesimerkissä lausejoukon Σ loogisten seurausten joukko on $\mathrm{Cn}(\Sigma) = \{\neg L \lor K, L \rightarrow P \land K, \neg P \lor P, \ldots\}$.

- Atomisten lauseiden joukkoon $\mathcal{P}$ perustuvia lausejoukkoja $\Sigma \subseteq \mathcal{L}$ voidaan olennaisesti määritellä $2^{2^{|\mathcal{P}|}}$ erilaista.

Tietämyksen esittäminen: ongelmana rajata mallien joukko sopivat lauseet valitsemalla siten, että saadaan halutut loogiset seuraukset.

- Lähtökohtana olevan tyhjän lausejoukon $\emptyset$ malleja ovat kaikki totuusjakelut. Täten $\mathrm{Cn}(\emptyset)$ on itse asiassa pätevien lauseiden joukko.
- Lauseiden lisääminen karsii mahdollisesti mallien joukkoa ja kasvattaa loogisten seurausten joukkoa *(monotonisuus)*.
- Toisessa ääripäässä lausejoukko Σ voi tulla *ristiriitaiseksi*, jolloin sillä ei ole yhtään mallia ja kaikki lauseet ovat tällöin lausejoukon loogisia seurauksia (eli $\mathrm{Cn}(\Sigma) = \mathcal{L}$).

Huomio. Kaikille lausejoukoille Σ pätee $\mathrm{Cn}(\emptyset) \subseteq \mathrm{Cn}(\Sigma)$.

Esimerkki. Lamppuesimerkissä $\Sigma \not\models L \leftrightarrow K$, mutta lauseen P lisäämisestä seuraa $\Sigma \cup \{P\} \models L \leftrightarrow K$. Sen sijaan lausejoukko $\Sigma \cup \{K \leftrightarrow \neg P,\ L\}$ on ristiriitainen/toteutumaton.

4 Semanttinen taulu

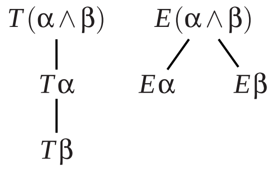
- Konnektiivikohtaiset taulusäännöt
- Semanttisen taulun määritelmä
- Taulumenetelmän ominaisuuksia
- Loogisten ongelmien ratkominen taulumenetelmällä
- Esimerkkejä

4.1 Konnektiivikohtaiset taulusäännöt

- Totuustaulukkoja käytetään lauseen ϕ totuusarvon laskemiseen, kun annettuna on atomisten lauseiden $A \in \mathrm{At}(\phi)$ totuusarvot.
- Semanttisissa tauluissa idea on käänteinen: määrätään lauseen ϕ alilauseiden (ja lopulta atomisten lauseiden $A \in \mathrm{At}(\phi)$) totuusarvoja lähtien liikkeelle lauseen ϕ totuusarvosta (tosi tai epätosi).

Esimerkki. Verrataan konjunktion totuustaulukkoa ja taulusääntöjä:

α	β	$(\alpha \land \beta)$
T	T	T
T	E	E
E	T	E
E	E	E



Konnektiivien ¬, ∨, ∧ ja → taulusäännöt ovat seuraavat:

$$\begin{array}{c} T\neg\alpha \\ | \\ E\alpha \end{array} \qquad \begin{array}{ccc} & T(\alpha\vee\beta) & \\ / & & \backslash \\ T\alpha & & T\beta \end{array} \qquad \begin{array}{c} T(\alpha\wedge\beta) \\ | \\ T\alpha \\ | \\ T\beta \end{array} \qquad \begin{array}{ccc} & T(\alpha\rightarrow\beta) & \\ / & & \backslash \\ E\alpha & & T\beta \end{array}$$

$$\begin{array}{c} E\neg\alpha \\ | \\ T\alpha \end{array} \qquad \begin{array}{c} E(\alpha\vee\beta) \\ | \\ E\alpha \\ | \\ E\beta \end{array} \qquad \begin{array}{ccc} & E(\alpha\wedge\beta) & \\ / & & \backslash \\ E\alpha & & E\beta \end{array} \qquad \begin{array}{c} E(\alpha\rightarrow\beta) \\ | \\ T\alpha \\ | \\ E\beta \end{array}$$

Ajatuksena on, että jokaiselle juurisolmusta lehtisolmuun johtavalle *polulle* kirjatut totuusarvovaatimukset ovat yhtäaikaa voimassa.

Konnektiivien $\underline{\vee}$ ja ↔ taulusäännöt ovat seuraavat:

$$\begin{array}{ccc} & T(\alpha\underline{\vee}\beta) & \\ / & & \backslash \\ T\alpha & & E\alpha \\ | & & | \\ E\beta & & T\beta \end{array} \qquad \begin{array}{ccc} & T(\alpha\leftrightarrow\beta) & \\ / & & \backslash \\ T\alpha & & E\alpha \\ | & & | \\ T\beta & & E\beta \end{array}$$

$$\begin{array}{ccc} & E(\alpha\underline{\vee}\beta) & \\ / & & \backslash \\ T\alpha & & E\alpha \\ | & & | \\ T\beta & & E\beta \end{array} \qquad \begin{array}{ccc} & E(\alpha\leftrightarrow\beta) & \\ / & & \backslash \\ T\alpha & & E\alpha \\ | & & | \\ E\beta & & T\beta \end{array}$$

Huomio. Näistä taulusäännöistä voi todeta konnektiivien $\underline{\vee}$ ja ↔ väliset suhteet: $\alpha\underline{\vee}\beta \equiv \alpha\leftrightarrow\neg\beta \equiv \neg\alpha\leftrightarrow\beta \equiv \neg(\alpha\leftrightarrow\beta)$.

Ketjukonjunktioille ja -disjunktioille voidaan johtaa seuraavat säännöt:

$$\begin{array}{c} T(\alpha_1\wedge\ldots\wedge\alpha_n) \\ | \\ T\alpha_1 \\ \vdots \\ T\alpha_n \end{array} \qquad \begin{array}{ccccc} & & T(\alpha_1\vee\ldots\vee\alpha_n) & & \\ & / & | & \backslash & \\ T\alpha_1 & & \ldots & & T\alpha_n \end{array}$$

$$\begin{array}{ccccc} & & E(\alpha_1\wedge\ldots\wedge\alpha_n) & & \\ & / & | & \backslash & \\ E\alpha_1 & & \ldots & & E\alpha_n \end{array} \qquad \begin{array}{c} E(\alpha_1\vee\ldots\vee\alpha_n) \\ | \\ E\alpha_1 \\ \vdots \\ E\alpha_n \end{array}$$

Huomio. Jatkossa nämä ymmärretään lyhennysmerkintöinä!

Taulusäännön instantiointi

Esimerkki. Tarkastellaan lausetta $(\neg A\rightarrow C\vee D)\rightarrow A\vee B$:

Lause on muodoltaan implikaatio $\alpha\rightarrow\beta$, missä ehtona on lause $\alpha = \neg A\rightarrow C\vee D$ ja seurauksena lause $\beta = A\vee B$.

Korvataan α ja β kyseisillä lauseilla implikaation säännössä:

$$\begin{array}{ccc} & T(\alpha\rightarrow\beta) & \\ / & & \backslash \\ E\alpha & & T\beta \end{array} \quad \leadsto \quad \begin{array}{ccc} & T((\neg A\rightarrow C\vee D)\rightarrow A\vee B) & \\ / & & \backslash \\ E(\neg A\rightarrow C\vee D) & & T(A\vee B) \end{array}$$

Huomio. Jatkossa on olennaista osata tunnistaa, mitä muotoa mikin lause on, koska taulusäännön valinta suoritetaan tällä perusteella.

4.2 Semanttisen taulun määritelmä

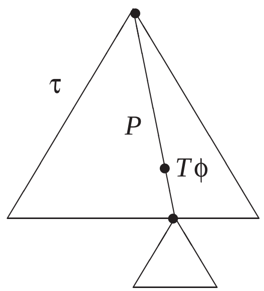
- Edellä määriteltiin jokaiselle lausetyypille omat taulusääntönsä.
- Yksittäinen taulusääntö tuottaa totuusarvovaatimuksia alilauseille lauseen totuusarvosta lähtien.
- Näitä vaatimuksia voidaan analysoida taulusäännöillä rekursiivisesti, kunnes saadaan atomisia lauseita koskevia totuusarvovaatimuksia.

Seuraavaksi määriteltävät *semanttiset taulut* ovat rakenteeltaan puita, joiden

1. solmujen asteluku on enintään kaksi ja
2. solmuina on muotoa $T\phi$ ja $E\phi$ olevia lausekkeita, jotka tulkitaan totuusarvovaatimuksiksi asianomaisille lauseille ϕ.

Määritelmä. *Semanttinen taulu* muodostetaan seuraavilla periaatteilla:

- Jokainen yksisolmuinen puu, jonka ainoana solmuna on juurisolmu $T\phi$ tai $E\phi$, on semanttinen taulu.
- Monimutkaisempia tauluja muodostetaan seuraavalla tavalla: Olkoon τ semanttinen taulu. Jos P on jokin τ:n polku juurisolmusta lehtisolmuun, $T\phi$ ($E\phi$) jokin τ:n solmu polulla P ja τ' saadaan τ:sta liittämällä $T\phi$:n ($E\phi$:n) taulusääntö *ilman juurisolmua* polun P jatkoksi, niin τ' on myöskin semanttinen taulu.



Esimerkki. Muodostetaan semanttinen taulu lähtien liikkeelle juurisolmusta $E(A \lor B \to A \land B)$.

1. $E(A \lor B \to A \land B)$
2. $T(A \lor B)^{1.}$
3. $E(A \land B)^{1.}$
4. $TA^{2.}$ | 4. $TB^{2.}$
5. $EA^{3.}$ | 5. $EB^{3.}$ | 5. $EA^{3.}$ | 5. $EB^{3.}$

Semanttisen taulun solmut on numeroitu luettavuuden parantamiseksi. *Poluille lisättävät solmut numeroidaan juoksevasti.* Yläindeksinä oleva numero kertoo, monennestako asianomaisen polun solmusta kyseinen solmu on saatu taulusääntöä soveltamalla.

Määritelmä. Olkoon τ semanttinen taulu, P polku juurisolmusta lehtisolmuun taulussa τ ja $T\phi$ ($E\phi$) polun P solmu.

1. Solmu $T\phi$ ($E\phi$) on *hajoitettu* polulla P, jos
   (a) ϕ on atominen lause, tai
   (b) solmun $T\phi$ ($E\phi$) taulusäännön jonkun polun P' jokainen solmu on polulla P.
2. Polku P on *ristiriitainen*, jos sillä esiintyy sekä $T\alpha$ että $E\alpha$ jollekin lauseelle α (tämän merkiksi kirjoitetaan usein $\times$ polun loppuun).
3. Polku P on *valmis*, jos se on ristiriitainen tai jokainen sen solmuista on hajoitettu polulla P.
4. Taulu τ on *valmis*, jos jokainen sen poluista on valmis.
5. Taulu τ on *ristiriitainen*, jos jokainen sen poluista on ristiriitainen.

Esimerkki. Palataan edelliseen esimerkkiin.

1. $E(A \vee B \rightarrow A \wedge B)$
2. $T(A \vee B)^{1.}$
3. $E(A \wedge B)^{1.}$

$4.\ TA^{2.}$ — $5.\ EA^{3.}$ $\times$, $5.\ EB^{3.}$

$4.\ TB^{2.}$ — $5.\ EA^{3.}$, $5.\ EB^{3.}$ $\times$

- Taulun reunimmaiset polut ovat ristiriitaiset.
- Kaikki taulun polut ovat valmiita.
- Taulu on valmis, muttei ristiriitainen.

4.3 Taulumenetelmän ominaisuuksia

Olkoon ϕ atomisten lauseiden joukkoon $\mathcal{P}$ perustuvan kielen $\mathcal{L}$ lause.

Semanttisella taululla voidaan ratkaista seuraava looginen ongelma: onko olemassa totuusjakelua $\mathcal{A} \subseteq \mathcal{P}$ siten, että $\mathcal{A} \models \phi$ ($\mathcal{A} \not\models \phi$)?

Todistamme jatkossa seuraavan ominaisuuden:

Väite. Olkoon $\phi \in \mathcal{L}$ mikä tahansa lause. Juurisolmusta $T\phi$ ($E\phi$) muodostetussa valmiissa semanttisessa taulussa on ei-ristiriitainen polku, jos ja vain jos $\mathcal{A} \models \phi$ ($\mathcal{A} \not\models \phi$) jollekin totuusjakelulle $\mathcal{A} \subseteq \mathcal{P}$.

Määritelmä. Semanttisen taulun τ polku P on *yhteensopiva* totuusjakelun $\mathcal{A}$ kanssa (merkitään $\mathcal{A} \| P$), jos ja vain jos

1. $\mathcal{A} \models \alpha$ jokaiselle polun P solmulle $T\alpha$, ja
2. $\mathcal{A} \not\models \beta$ jokaiselle polun P solmulle $E\beta$.

Taulumenetelmän virheettömyys

Olkoon $\phi \in \mathcal{L}$ mikä tahansa lause ja τ juurisolmusta $T\phi$ ($E\phi$) muodostettu *valmis* semanttinen taulu.

Väite. *Jos* semanttisessa taulussa τ on ei-ristiriitainen polku P, *niin* $\mathcal{A} \| P$ ja $\mathcal{A} \models \phi$ ($\mathcal{A} \not\models \phi$) jokaiselle totuusjakelulle $\mathcal{A} \subseteq \mathcal{P}$ joka täyttää seuraavat vaatimukset kaikille atomisille lauseille $A \in \mathcal{P}$:

- jos TA esiintyy polulla P, niin $A \in \mathcal{A}$, ja
- jos EA esiintyy polulla P, niin $A \notin \mathcal{A}$.

Todistus. Olkoon P mikä tahansa taulun τ ei-ristiriitainen polku ja $\mathcal{A}$ totuusjakelu, joka täyttää edellä mainitut vaatimukset atomisille lauseille.

Polku P on valmis, koska taulu τ on valmis.

Todistetaan $\mathcal{A} \| P$ induktiolla polun P solmujen rakenteen suhteen.

Perustapaus: $TA \in P$ ($EA \in P$), missä A on $\mathcal{P}$:n atominen lause. Totuusjakelulle $\mathcal{A}$ asetettujen ehtojen perusteella $\mathcal{A} \models A$ ($\mathcal{A} \not\models A$).

Induktioaskel: Käydään läpi kaikki tapaukset $T(\alpha \wedge \beta) \in P$, $E(\alpha \wedge \beta) \in P$, $T(\alpha \vee \beta) \in P$, $E(\alpha \vee \beta) \in P$, $T(\alpha \rightarrow \beta) \in P$, ...

- $E(\alpha \rightarrow \beta) \in P$, jolloin $T\alpha \in P$ ja $E\beta \in P$, koska P on valmis.
  Induktio-oletuksella $\mathcal{A} \models \alpha$ ja $\mathcal{A} \not\models \beta$.
  Totuusmääritelmän perusteella tällöin $\mathcal{A} \not\models \alpha \rightarrow \beta$.
- $T(\alpha \rightarrow \beta) \in P$, jolloin $E\alpha \in P$ tai $T\beta \in P$, koska P on valmis.
  Induktio-oletuksella $\mathcal{A} \not\models \alpha$ jos $E\alpha \in P$, tai $\mathcal{A} \models \beta$ jos $T\beta \in P$.
  Totuusmääritelmän perusteella tällöin $\mathcal{A} \models \alpha \rightarrow \beta$.
- Muut tapaukset käsitellään samaan tapaan.

Koska $\mathcal{A} \| P$ ja $T\phi \in P$ ($E\phi \in P$), niin $\mathcal{A} \models \phi$ ($\mathcal{A} \not\models \phi$).

Esimerkki. Tarkastellaan joukkoon $\mathcal{P} = \{A, B\}$ perustuvan kielen $\mathcal{L}$ lausetta $(A \wedge B) \wedge (A \rightarrow B)$ sekä seuraavaa valmista semanttista taulua:

$$
\begin{array}{ccc}
 & E((A \wedge B) \wedge (A \rightarrow B)) & \\
E(A \wedge B) & & E(A \rightarrow B) \\
EA \quad EB & & TA \\
 & & | \\
 & & EB
\end{array}
$$

- Ei-ristiriitaisia polkuja on kolme P_1, P_2 ja P_3 (vasemmalta oikealle).
- Sama totuusjakelu voi olla yhteensopiva usean ei-ristiriitaisen polun kanssa. Esimerkissä totuusjakelulle $\mathcal{A}_1 = \emptyset$ pätee $\mathcal{A}_1 \| P_1$ ja $\mathcal{A}_1 \| P_2$.
- Yhteensopivia totuusjakeluja voi olla myös useita ($\mathcal{A}_1 \| P_2$ ja $\mathcal{A}_2 \| P_2$, missä $\mathcal{A}_2 = \{A\}$) tai näitä voi olla myös täsmälleen yksi ($\mathcal{A}_2 \| P_3$).

Taulumenetelmän täydellisyys

Olkoon $\phi \in \mathcal{L}$ mikä tahansa lause ja τ juurisolmusta $T\phi$ ($E\phi$) muodostettu (*ei välttämättä valmis*) semanttinen taulu.

Väite. *Jos* $\mathcal{A} \models \phi$ ($\mathcal{A} \not\models \phi$) jossain totuusjakelussa $\mathcal{A} \subseteq \mathcal{P}$, *niin* semanttisessa taulussa τ on ei-ristiriitainen polku P siten, että $\mathcal{A} \| P$.

Todistus. Oletetaan $\mathcal{A} \models \phi$ ($\mathcal{A} \not\models \phi$) jollekin totuusjakelulle $\mathcal{A} \subseteq \mathcal{P}$. Käytetään induktiota semanttisen taulun τ rakenteen suhteen (vrt. semanttisen taulun induktiivinen määritelmä).

Perustapaus: taulu τ koostuu ainoastaan juurisolmusta $T\phi$ ($E\phi$). Tällöin ainoa mahdollinen polku P juurisolmusta $T\phi$ ($E\phi$) itseensä toteuttaa vaatimuksen $\mathcal{A} \| P$, koska $T\phi$ ($E\phi$) on polun ainoa solmu.

Induktioaskel: taulu τ saadaan taulusta τ' soveltamalla taulusääntöä τ':n polun P solmuun $T\gamma$ tai $E\gamma$ (jollekin lauseelle γ). Induktio-oletuksen nojalla τ':ssa polku P' siten, että $\mathcal{A} \| P'$.

Jos $P \neq P'$, niin P' myös taulun τ polku.
Jos $P = P'$, niin $\mathcal{A} \| P$ ja päädymme tapausanalyysiin:

- $\gamma = \neg\alpha$ ja $T(\neg\alpha) \in P$:
  $\Longrightarrow$ $\mathcal{A} \models \neg\alpha$ (koska $\mathcal{A} \| P$) ja $E\alpha$ lisätään P:n jatkoksi
  $\Longrightarrow$ $\mathcal{A} \not\models \alpha$ ja syntyvä τ:n polku on yhteensopiva $\mathcal{A}$:n kanssa.
- $\gamma = \neg\alpha$ ja $E(\neg\alpha) \in P$:
  $\Longrightarrow$ $\mathcal{A} \not\models \neg\alpha$ (koska $\mathcal{A} \| P$) ja solmu $T\alpha$ lisätään P:n jatkoksi.
  $\Longrightarrow$ $\mathcal{A} \models \alpha$ ja syntyvä τ:n polku on yhteensopiva $\mathcal{A}$:n kanssa.
- Muut tapaukset (vaihtoehdot lauseeksi γ) käsitellään vastaavasti.

Edellä osoitettu taulumenetelmän ominaisuus voidaan todeta seuraavasta esimerkistä.

Esimerkki. Totuusjakelu $\mathcal{A} = \{B\}$ on yhteensopiva allaolevan semanttisen taulun oikeanpuoleisen polun kanssa.

$$
\begin{array}{c}
E((A \vee B) \rightarrow A) \\
| \\
T(A \vee B) \\
| \\
EA \\
TA \qquad TB \\
\times \qquad\quad
\end{array}
$$

Täten $\mathcal{A} \not\models (A \vee B) \rightarrow A$, $\mathcal{A} \models A \vee B$, $\mathcal{A} \not\models A$ ja $\mathcal{A} \models B$.

4.4 Loogisten ongelmien ratkominen taulumenetelmällä

Lauseen toteutuvuuden tutkiminen

- Lause $\phi \in \mathcal{L}$ on *toteutuva*

  $\iff \exists$ totuusjakelu $\mathcal{A} \subseteq \mathcal{P}$ s.e. $\mathcal{A} \models \phi$

  $\iff$ juurisolmusta $T\phi$ muodostetussa valmiissa semanttisessa taulussa on ainakin yksi ei-ristiriitainen polku P.

- Jokaisesta ei-ristiriitaisesta polusta P voidaan konstruoida lauseelle ϕ malleja $\mathcal{A}$ seuraavasti. Jos atomiselle lauseelle $A \in \mathcal{P}$ pätee
  - $TA \in P$, niin $A \in \mathcal{A}$
  - $EA \in P$, niin $A \notin \mathcal{A}$
  - $TA \notin P$ ja $EA \notin P$, niin voidaan valita joko $A \in \mathcal{A}$ tai $A \notin \mathcal{A}$.

Esimerkki. Tutkitaan lauseen $A \vee B \rightarrow A$ toteutuvuutta.

1. $T((A \vee B) \rightarrow A)$

2. $E(A \vee B)^{1.}$ 2. $TA^{1.}$

3. $EA^{2.}$

4. $EB^{2.}$

Ei-ristiriitaisista poluista voidaan muodostaa lauseelle mallit $\mathcal{A}_1 = \{\}$, $\mathcal{A}_2 = \{A\}$ ja $\mathcal{A}_3 = \{A, B\}$.

Huomio. Mallien lukumäärä voi kasvaa eksponentiaalisesti lauseen pituuteen nähden (esim. lause $(A_1 \vee B_1) \wedge \ldots \wedge (A_n \vee B_n)$), joten kaikkien mallien ylöskirjaaminen ei välttämättä ole tilasyistä mielekästä.

Lausejoukon toteutuvuuden tutkiminen

- Äärellinen lausejoukko $\Sigma = \{\phi_1, \ldots, \phi_n\} \subseteq \mathcal{L}$ on toteutuva

  $\iff$ lause $\phi_1 \wedge \cdots \wedge \phi_n \in \mathcal{L}$ on toteutuva

  $\iff$ juurisolmusta $T(\phi_1 \wedge \cdots \wedge \phi_n)$ muodostetussa valmiissa semanttisessa taulussa on ainakin yksi ei-ristiriitainen polku P.

- Lausejoukolle voidaan konstruoida malleja samaan tapaan kuin yksittäisen lauseen tapauksessa.

- Käytännössä juurisolmu $T(\phi_1 \wedge \cdots \wedge \phi_n)$ voidaan jättää kirjoittamatta näkyviin tilan säästämiseksi ja taulun juureen voidaan kirjata samalle polulle solmut $T\phi_1, \ldots, T\phi_n$.

Lauseen pätevyyden tutkiminen

- $\models \phi$

  $\iff \neg\phi$ on toteutumaton

  $\iff$ juurisolmusta $T\neg\phi$ muodostettu valmis semanttinen taulu on ristiriitainen

  $\iff$ juurisolmusta $E\phi$ muodostettu valmis semanttinen taulu on ristiriitainen.

Täten taulumenetelmästä saadaan lauselogiikalle todistusmenetelmä.

Määritelmä. Taulu τ on lauseen ϕ *todistus*, jos taulun τ juurisolmuna on $E\phi$ ja τ on ristiriitainen (ja siten myös valmis).

Jos lauseella ϕ on todistus, ϕ:tä sanotaan *teoreemaksi/todistuvaksi* (merk. $\vdash \phi$).

Esimerkki. Osoitetaan lause $A \wedge B \rightarrow A \vee B$ teoreemaksi:

1. $E(A \wedge B \rightarrow A \vee B)$
2. $T(A \wedge B)^{1.}$
3. $E(A \vee B)^{1.}$
4. $TA^{2.}$
5. $TB^{2.}$
6. $EA^{3.}$
7. $EB^{3.}$

$\times$

Todistusmenetelmän virheettömyys ja täydellisyys

Todistusmenetelmä M on

- *virheetön*, jos lauseen ϕ todistettavuudesta menetelmällä M ($\vdash_M \phi$) seuraa lauseen ϕ pätevyys ($\models \phi$).
- *täydellinen*, jos lauseen ϕ pätevyydestä ($\models \phi$) seuraa lauseen todistettavuus menetelmällä M ($\vdash_M \phi$)

Edellä määritelty semanttiseen tauluun perustuva todistusmenetelmä on virheetön ja täydellinen, koska $\vdash \phi \iff \models \phi$.

Huomio. Semanttisiin tauluihin perustuva menetelmä poikkeaa melkoisesti klassisista todistusjärjestelmistä, joissa pätevät lauseet tuotetaan syntaktisesti aksiomien ja päättelysääntöjen avulla.

Loogisen seuraavuuden tutkiminen

- $\{\phi_1, \ldots, \phi_n\} \models \phi$
  $\iff \models \phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi$
  $\iff$ juurisolmusta $E(\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi)$ muodostettu valmis semanttinen taulu on ristiriitainen.
- Jos juurisolmusta $E(\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi)$ muodostettuun valmiiseen semanttiseen tauluun jää ei-ristiriitaisia polkuja, näistä voidaan konstruoida vastamalleja $\mathcal{A}$ siten, että $\mathcal{A} \models \{\phi_1, \ldots, \phi_n\}$ ja $\mathcal{A} \not\models \phi$.
- Käytännössä taulun juureen tulevat solmut $E(\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi)$ ja $T(\phi_1 \wedge \cdots \wedge \phi_n)$ voidaan jättää kirjoittamatta näkyviin, ja taulu aloitetaan totuusarvovaatimuksilla $E\phi$ ja $T\phi_1$,, ..., $T\phi_n$ kirjattuina samalle polulle. Tämä voidaan nähdä vastamallin spesifikaationa.

Johdettavuus lausejoukosta

Määritelmä. Lause ϕ on johdettavissa äärellisestä lausejoukosta $\Sigma = \{\phi_1, \ldots, \phi_n\}$ (merkitään $\Sigma \vdash \phi$), joss juurisolmusta $E(\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi)$ muodostettu valmis semanttinen taulu on ristiriitainen.

Huomioita.

- $\Sigma \vdash \phi \iff \Sigma \models \phi$.
- Semanttisiin tauluihin perustuva todistusmenetelmä on täten virheetön ja täydellinen myös tutkittaessa loogista seuraavuutta.

Loogisen ekvivalenssin tutkiminen

- Lauseet ϕ ja ψ ovat loogisesti ekvivalentit
  $\iff \models \phi \leftrightarrow \psi$
  $\iff$ juurisolmusta $E(\phi \leftrightarrow \psi)$ muodostettu valmis semanttinen taulu on ristiriitainen.
- Äärellisten lausejoukkojen Σ_1 ja Σ_2 loogisen ekvivalenssin toteaminen edellyttää mahdollisesti usean semanttisen taulun konstruointia:
  1. Jokaiselle $\sigma_1 \in \Sigma_1$ tulee osoittaa $\Sigma_2 \models \sigma_1$.
  2. Jokaiselle $\sigma_2 \in \Sigma_2$ tulee osoittaa $\Sigma_1 \models \sigma_2$.

Esimerkki. Totea lamppuesimerkissä laadittujen spesifikaatioiden $\Sigma = \{\neg P \rightarrow \neg L, P \rightarrow (L \leftrightarrow K)\}$ ja $\Sigma' = \{L \rightarrow P, P \land L \rightarrow K, P \land K \rightarrow L\}$ välinen ekvivalenssi käyttämällä semanttista taulua!

Ohjeita semanttisten taulujen laadintaan:

- Taulun solmussa $T\phi$ (tai $E\phi$) olevan lauseen ϕ rakenne määrää, mitä taulusääntöä käytetään: esim. implikaation $\phi = \alpha \rightarrow \beta$ käsittely.
- Polun laajentaminen voidaan lopettaa ristiriidan ilmentymiseen.
- Solmujen hajoittamisjärjestyksellä voi usein vaikuttaa taulun kokoon.
- Taulun koon kasvua kannattaa välttää esim. seuraavilla periaatteilla.
  1. Hajoitetaan ensisijaisesti solmuja, jotka eivät haarauta taulua.
  2. Hajoitetaan toissijaisesti jäljelle jääviä solmuja, joista syntyvät totuusarvovaatimukset johtavat (välittömään) ristiriitaan.
  3. Tämän jälkeen hajoitetaan muita (taulua haarauttavia) solmuja.
- Kaikkia lauseita koskevat totuusarvovaatimukset eivät välttämättä ole tarpeen ristiriidan aikaansaamiseen (edes tietyllä polulla)!

Esimerkki. Tarkastellaan alla annettua polkua eräässä keskeneräisessä semanttisessa taulussa:

$$\begin{array}{c} \vdots \\ | \\ 5.\ TA \\ | \\ 6.\ E(B \rightarrow C) \\ | \\ 7.\ T(A \lor B) \\ | \\ 8.\ E(A \land C) \end{array}$$

- Solmun 6 käsittely lienee paras vaihtoehto (taulu ei haaraudu).
- Solmu 8 olisi seuraavaksi paras vaihtoehto, koska syntyvistä poluista toinen (jolle kirjataan solmu EA) on suoraan ristiriitainen.
- Solmun 7 käsittely haarauttaa taulun.

4.5 Esimerkkejä

Boolen funktioiden ja lauselogiikan yhteys

Mikä tahansa Boolen funktio $f : \{0,1\}^n \rightarrow \{0,1\}$ voidaan esittää lauselogiikan lauseena:

- Arvoja 0 ja 1 vastaavat totuusarvot E ja T
- Boolen muuttuja a (jolla arvona 0 tai 1) esitetään atomisena lauseena A (jolla totuusarvo E tai T)
- Komplementti esitetään negaation $\neg$ avulla
- Tulo $\cdot$ ja summa $+$ esitetään konjunktion $\land$ ja disjunktion $\lor$ avulla.

Esimerkki. Funktiolle $f(a,b,c) = a \cdot \overline{b} + c$ saadaan näillä periaatteilla esitys $(A \land \neg B) \lor C$.

Väite. Olkoon f boolen funktio ja lause ϕ sitä vastaava esitys:

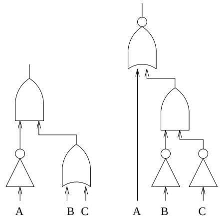
funktion f arvon laskeminen tietyillä muuttujien arvoilla vastaa lauseen ϕ totuusarvon laskentaa vastaavassa totuusjakelussa.

Esimerkki.
Jos $a = 0$, $b = 0$ ja $c = 1$, niin vastaava totuusjakelu on $\mathcal{A} = \{C\}$.

Aikaisemmalle esimerkille $f(0,0,1) = 0 \cdot \overline{0} + 1 = 0 \cdot 1 + 1 = 0 + 1 = 1$ ja vastaavasti $\mathcal{A} \models (A \wedge \neg B) \vee C$.

Esimerkki.

Jos haluamme osoittaa, että kaksi Boolen funktiota f ja f' ovat samat, riittää, että todeamme lause-esitysten ϕ ja ϕ' loogisen ekvivalenssin ($\phi \equiv \phi' \iff \models \phi \leftrightarrow \phi'$).

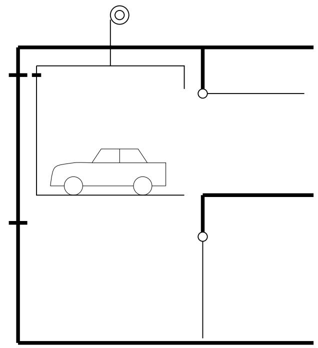


1. $E(\neg A \wedge (B \vee C) \leftrightarrow \neg(A \vee (\neg B \wedge \neg C)))$

Vasen haara:
2. $T(\neg A \wedge (B \vee C))$
3. $E\neg(A \vee (\neg B \wedge \neg C))$
4. $T\neg A$
5. $T(B \vee C)$
6. EA
7. $T(A \vee (\neg B \wedge \neg C))$
- 8. TA $\times$
- 8. $T(\neg B \wedge \neg C)$
  9. $T\neg B$
  10. $T\neg C$
  11. EB
  12. EC
  - 13. TB $\times$
  - 13. TC $\times$

Oikea haara:
2. $E(\neg A \wedge (B \vee C))$
3. $T\neg(A \vee (\neg B \wedge \neg C))$
4. $E(A \vee (\neg B \wedge \neg C))$
5. EA
6. $E(\neg B \wedge \neg C)$
- 7. $E\neg A$
  8. TA $\times$
- 7. $E(B \vee C)$
  8. EB
  9. EC
  - 10. $E\neg B$
    11. TB $\times$
  - 10. $E\neg C$
    11. TC $\times$

Esimerkki. Mallinnetaan yksinkertaistettua hissijärjestelmää lauselogiikalla:



Atomiset lauseet:

A_i: kerroksen i ovi on auki ja
K_i: hissi on kerroksessa i,

missä $i = 1$ tai $i = 2$.

Kuvataan lauselogiikalla järjestelmän sallitut tilat (eli haetaan lausejoukko Σ, jonka *mallit* vastaavat sallittuja tiloja).

- Tyhjällä lausejoukolla $\emptyset$ on $2^4 = 16$ mallia, kuten esimerkiksi $\mathcal{A} = \{K_1, K_2, A_1, A_2\}$.
  Tämä vastaa järjestelmän tilaa, jossa hissi on yhtäaikaa molemmissa kerroksissa ja molemmat ovet ovat auki. Tämän pitäisi olla mahdotonta, joten tarvitaan lisää lauseita (rajoitteita).
- Fyysisen maailman asettama rajoitus: $\neg K_1 \vee \neg K_2$
  (huomaa, että edellä annettu $\mathcal{A}$ ei ole tämän lauseen malli).
- Emme kuitenkaan halua asettaa vaatimusta $K_1 \vee K_2$
  (hissi voi olla matkalla kerrosten välillä).
- Oville asetettavat turvallisuusvaatimukset: $A_1 \rightarrow K_1$ ja $A_2 \rightarrow K_2$.

Järjestelmälle saadaan spesifikaatioksi lausejoukko

$$\Sigma = \{\neg K_1 \vee \neg K_2, A_1 \rightarrow K_1, A_2 \rightarrow K_2\}.$$

Etsitään spesifikaation mallit semanttisen taulun avulla.

1. $T(\neg K_1 \vee \neg K_2)$
2. $T(A_1 \rightarrow K_1)$
3. $T(A_2 \rightarrow K_2)$
   - 4. $T\neg K_1\,^{1.}$
     - 5. $EK_1\,^{4.}$
       - 6. $EA_1\,^{2.}$
         - 7. $EA_2\,^{3.}$
         - 7. $TK_2\,^{3.}$
       - 6. $TK_1\,^{2.}$ $\times$
   - 4. $T\neg K_2\,^{1.}$
     - 5. $EK_2\,^{4.}$
       - 6. $EA_2\,^{3.}$
         - 7. $EA_1\,^{2.}$
         - 7. $TK_1\,^{2.}$
       - 6. $TK_2\,^{3.}$ $\times$

Taulun ei-ristiriitaisista haaroista voidaan muodostaa lausejoukolle viisi erilaista mallia, jotka vastaavat järjestelmän sallittuja tiloja:

$$\mathcal{A}_1 = \emptyset,\ \mathcal{A}_2 = \{K_1\},\ \mathcal{A}_3 = \{K_1, A_1\},\ \mathcal{A}_4 = \{K_2\} \text{ ja } \mathcal{A}_5 = \{K_2, A_2\}.$$

Voimme myös osoittaa, että hissin molemmat ovet eivät voi olla yhtäaikaisesti auki (turvallisuusominaisuus):

$$\Sigma \models \neg(A_1 \wedge A_2).$$

1. $E\neg(A_1 \wedge A_2)$
2. $T(\neg K_1 \vee \neg K_2)$
3. $T(A_1 \rightarrow K_1)$
4. $T(A_2 \rightarrow K_2)$
5. $T(A_1 \wedge A_2)\,^{1.}$
6. $TA_1\,^{5.}$
7. $TA_2\,^{5.}$
   - 8.$EA_1\,^{3.}$ $\times$
   - 8.$TK_1\,^{3.}$
     - 9.$EA_2\,^{4.}$ $\times$
     - 9.$TK_2\,^{4.}$
       - 10.$T\neg K_1\,^{2.}$
         - 11.$EK_1\,^{10.}$ $\times$
       - 10.$T\neg K_2\,^{2.}$
         - 11.$EK_2\,^{10.}$ $\times$

5 Vaihtoehtoisia todistusjärjestelmiä

- Hilbertin järjestelmä
- Suppesin järjestelmä
- Järjestelmien välistä vertailua

5.1 Hilbertin järjestelmä

Hilbertin järjestelmä perustuu seuraaviin *aksiomiin* ja *päättelysääntöön*:

Aksiomat:

1. $\alpha \rightarrow (\beta \rightarrow \alpha)$
2. $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
3. $(\neg\beta \rightarrow \neg\alpha) \rightarrow ((\neg\beta \rightarrow \alpha) \rightarrow \beta)$

Päättelysääntö:

$$\frac{\alpha \rightarrow \beta \qquad \alpha}{\beta} \quad \text{(modus ponens)}$$

- Ajatuksena on, että pätevät lauseet pystytään tuottamaan *syntaktisesti* aksiomien instansseina (valisemalla α, β ja γ sopivasti) tai modus ponens -päättelysääntöä soveltamalla.
- Hilbertin järjestelmässä negaatio ja implikaatio ovat peruskonnektiiveina.

Huomio. Muut lauselogiikan peruskonnektiivit ovat lausuttavissa negaation ja implikaation avulla seuraavasti:

$\alpha \vee \beta \equiv \neg\neg\alpha \vee \beta \equiv \neg\alpha \rightarrow \beta$,

$\alpha \wedge \beta \equiv \neg\neg(\alpha \wedge \beta) \equiv \neg(\neg\alpha \vee \neg\beta) \equiv \neg(\alpha \rightarrow \neg\beta)$ ja

$\alpha \leftrightarrow \beta \equiv (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha) \equiv \neg((\alpha \rightarrow \beta) \rightarrow \neg(\beta \rightarrow \alpha))$.

Määritelmä. Olkoon Σ joukko lauseita.

Todistus lausejoukosta Σ on jono lauseita $\alpha_1, \ldots, \alpha_n$ siten, että kaikille $i \in \{1, \ldots, n\}$ pätee jokin seuraavista

- $\alpha_i \in \Sigma$,
- α_i on aksioman instanssi tai
- α_i on saatu modus ponensilla lauseista α_j, missä $j < i$ ja $\alpha_k = (\alpha_j \rightarrow \alpha_i)$, missä $k < i$.

Määritelmä. Lause α on *johdettavissa* lausejoukosta Σ Hilbert-järjestelmällä (merk. $\Sigma \vdash_{\mathrm{H}} \alpha$), joss on olemassa todistus $\alpha_1, \ldots, \alpha_n$ lausejoukosta Σ siten, että $\alpha = \alpha_n$.

Määritelmä. Lause α on *teoreema/todistuva* Hilbert-järjestelmässä (merk. $\vdash_{\mathrm{H}} \alpha$), joss $\emptyset \vdash_{\mathrm{H}} \alpha$

Esimerkki. Osoitetaan $\{A, B \rightarrow (A \rightarrow C)\} \vdash_{\mathrm{H}} B \rightarrow C$.

1.	A	olettamus
2.	$B \rightarrow (A \rightarrow C)$	olettamus
3.	$A \rightarrow (B \rightarrow A)$	aksioma 1
4.	$B \rightarrow A$	MP, 1, 3
5.	$(B \rightarrow (A \rightarrow C)) \rightarrow ((B \rightarrow A) \rightarrow (B \rightarrow C))$	aksioma 2
6.	$(B \rightarrow A) \rightarrow (B \rightarrow C)$	MP, 2, 5
7.	$B \rightarrow C$	MP, 4, 6

Väite. Olkoon lausejono $\alpha_1, \ldots, \alpha_n$ todistus lausejoukosta Σ Hilbertin järjestelmässä. Tällöin $\Sigma \models \alpha_i$ kaikille $i \in \{1, \ldots, n\}$.

Todistus. Käytetään induktiota i:n suhteen.

- **Perustapaus:** $i = 1$
  1. α_1 on instanssi jostain kolmesta aksiomasta
     $\Longrightarrow \models \alpha_1$ (voidaan osoittaa esim. semanttisilla tauluilla)
     $\Longrightarrow \Sigma \models \alpha_1$ (loogisen seurausrelaation ominaisuudet).
  2. $\alpha_1 \in \Sigma$
     $\Longrightarrow \Sigma \models \alpha_1$ (loogisen seurausrelaation ominaisuudet).
- **Induktioaskel:** $i > 1$

  Tapaukset missä lausejonon jäsen α_i on jonkin aksioman instanssi tai $\alpha_i \in \Sigma$ käsitellään kuten edellä.

- **Induktioaskel jatkuu:**

  Jos α_i saatiin modus ponensilla lauseista α_j ja $\alpha_k = (\alpha_j \rightarrow \alpha_i)$, missä $j < i$ ja $k < i$, niin induktiohypoteesin nojalla saadaan $\Sigma \models \alpha_j$ ja $\Sigma \models \alpha_j \rightarrow \alpha_i$.

  Oletetaan $\Sigma \not\models \alpha_i$

  $\Longrightarrow$ $\exists$ totuusjakelu $\mathcal{A}$ s.e. $\mathcal{A} \models \Sigma$ ja $\mathcal{A} \not\models \alpha_i$

  $\Longrightarrow$ $\mathcal{A} \not\models \alpha_j$ (koska $\Sigma \models \alpha_j \rightarrow \alpha_i \Longrightarrow \mathcal{A} \models \alpha_j \rightarrow \alpha_i$)

  $\Longrightarrow$ $\Sigma \not\models \alpha_j$, ristiriita.

  Siis $\Sigma \models \alpha_i$.

Hilbertin järjestelmä on virheetön ja täydellinen.

Väite. Jos $\Sigma \vdash_{\mathrm{H}} \alpha$, niin $\Sigma \models \alpha$ (todistettiin edellä).
Jos $\Sigma \models \alpha$, niin $\Sigma \vdash_{\mathrm{H}} \alpha$ (todistus sivuutetaan).

5.2 Suppesin järjestelmä

Suppesin luonnollisen päättelyn järjestelmässä ei ole aksiomia ja päättelysäännöt ovat seuraavat:

MP (modus ponens):

$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

TT (modus tollendo tollens):

$$\frac{\alpha \rightarrow \beta \quad \neg\beta}{\neg\alpha}$$

TP (modus tollendo ponens):

$$\frac{\alpha \vee \beta \quad \neg\alpha}{\beta}$$

Vaihtosäännöt:

KV:

$$\frac{\alpha \wedge \beta}{\beta \wedge \alpha}$$

DV:

$$\frac{\alpha \vee \beta}{\beta \vee \alpha}$$

Tuonti- ja eliminointisäännöt:

KNT:

$$\frac{\alpha}{\neg\neg\alpha}$$

KT:

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta}$$

DT:

$$\frac{\alpha}{\alpha \vee \beta \quad \beta \vee \alpha}$$

ET:

$$\frac{\alpha \rightarrow \beta \quad \beta \rightarrow \alpha}{\alpha \leftrightarrow \beta}$$

KNE:

$$\frac{\neg\neg\alpha}{\alpha}$$

KE:

$$\frac{\alpha \wedge \beta}{\alpha \quad \beta}$$

DE:

$$\frac{\alpha \vee \alpha}{\alpha}$$

EE:

$$\frac{\alpha \leftrightarrow \beta}{\alpha \rightarrow \beta \quad \beta \rightarrow \alpha}$$

Huomio. Implikaation eliminointi tapahtuu päättelysäännöillä MP, TT tai ES ja tuonti päättelysäännöillä ET ja HS (kts. myös seuraava kalvo).

DM 1:

$$\frac{\alpha \wedge \beta}{\neg(\neg\alpha \vee \neg\beta)}$$

DM 2:

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta}$$

DM 3:

$$\frac{\alpha \vee \beta}{\neg(\neg\alpha \wedge \neg\beta)}$$

DM 4:

$$\frac{\neg(\alpha \vee \beta)}{\neg\alpha \wedge \neg\beta}$$

HS:

$$\frac{\alpha \rightarrow \beta \quad \beta \rightarrow \gamma}{\alpha \rightarrow \gamma}$$

DS:

$$\frac{\alpha \vee \beta \quad \alpha \rightarrow \gamma \quad \beta \rightarrow \delta}{\gamma \vee \delta}$$

ET (ehdollinen todistaminen):

$$\begin{array}{c} [\alpha]^{(1)} \\ \vdots \\ \beta \\ \hline \alpha \rightarrow \beta \end{array}$$

ES (epäsuora todistaminen):

$$\frac{\neg\beta \rightarrow \alpha \wedge \neg\alpha}{\beta}$$

(1) Apuolettamus α merkitään hakasuluilla perutuksi, kun johtopäätös $\alpha \rightarrow \beta$ on tehty.

Määritelmä. Suppes-todistus lausejoukosta Σ on jono lauseita $\alpha_1,\ldots,\alpha_n$, joihin liittyy apuolettamusten joukot $H_0=\emptyset$ ja $H_1,\ldots,H_n$ siten, että kaikille $i\in\{1,\ldots,n\}$ pätee jokin seuraavista:

- $H_i=H_{i-1}$ ja $\alpha_i\in\Sigma$.
- $H_i=H_{i-1}\cup\{\alpha_i\}$ ja α_i on uusi apuolettamus $\alpha_i\notin H_{i-1}$.
- $H_i=H_{i-1}$ ja α_i on saatu jollain Suppes-järjestelmän päättelysäännöllä (paitsi ehdollisen todistamisen säännöllä) jonon aikaisemmista lauseista α_j $(j<i)$, joille $H_j\subseteq H_{i-1}$.
- $H_i=H_{i-1}$ ja $\alpha_i=\alpha_j$ jollekin jonon aikaisemmalle lauseelle α_j $(j<i)$, jolle $H_j\subseteq H_{i-1}$.
- $H_i=H_{i-1}-\{\alpha_j\}$ ja $\alpha_i=\alpha_j\rightarrow\alpha_{i-1}$ on saatu ehdollisen todistamisen säännöllä viimeisimmästä apuolettamuksesta α_j $(j<i)$ ja edeltävästä lauseesta α_{i-1}.

Määritelmä.

1. Lause α on *johdettavissa* Suppes-järjestelmällä lausejoukosta Σ (merk. $\Sigma\vdash_S\alpha$), joss on olemassa Suppes-todistus $\alpha_1,\ldots,\alpha_n$ lausejoukosta Σ siten, että $\alpha_n=\alpha$ ja $H_n=\emptyset$.
2. Lause α on *teoreema/todistuva* Suppes-järjestelmässä (merk. $\vdash_S\alpha$), joss $\emptyset\vdash_S\alpha$

Huomio. Suppes-todistukselle asetettu lisäehto $H_n=\emptyset$ edellyttää, että kaikki apuolettamukset on peruttava johtopäätöksen tekoon mennessä.

Väite. Suppesin järjestelmä on lauselogiikan virheetön ja täydellinen todistusmenetelmä $(\Sigma\vdash_S\alpha\iff\Sigma\models\alpha)$.

Esimerkki.

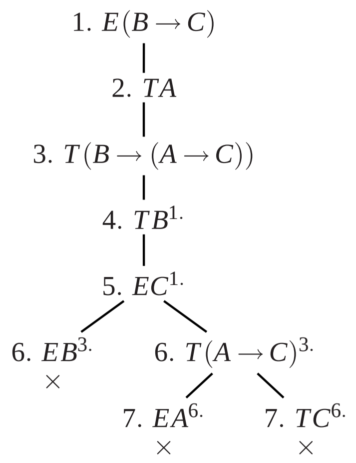
Osoitetaan $\{A, B\rightarrow(A\rightarrow C)\}\vdash_S B\rightarrow C$.

1.	A	olettamus	$H_1=\emptyset$
2.	$B\rightarrow(A\rightarrow C)$	olettamus	$H_2=\emptyset$
3.	B	apuolettamus	$H_3=\{B\}$
4.	$A\rightarrow C$	MP, 3, 2	$H_4=\{B\}$
5.	C	MP, 1, 4	$H_5=\{B\}$
6.	$B\rightarrow C$	ET, 3, 5	$H_6=\emptyset$

Käytännössä apuolettamusten joukoista pidetään kirjaa tekemällä todistukseen sisennyksiä, jolloin H_i:t voidaan jättää pois todistuksesta.

5.3 Järjestelmien vertailua

Esimerkki.
Osoitetaan vielä semanttisella taululla, että $B\rightarrow C$ on johdettavissa lausejoukosta $\{A, B\rightarrow(A\rightarrow C)\}$.



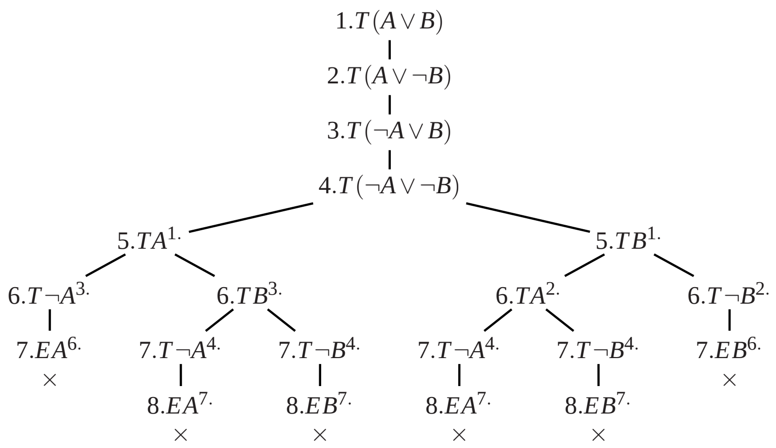
1. **Hilbertin järjestelmä**
   + Minimalistinen koneisto teoreemien tuottamiseksi.
   – Lauseet esitettävä implikaation ja negaation avulla.
   – Yksittäisten todistusten löytäminen voi olla vaikeaa.
   – Hankala todeta milloin lause ei ole todistuva/johdettavissa.
2. **Suppesin järjestelmä**
   + Päättelysääntöjen laaja valikoima tukee erilaisten loogisten päätelmien tekemistä ja todistusten löytämistä.
   + Lauseissa voi esiintyä kaikkia peruskonnektiiveja.
   – Hankala todeta milloin lause ei ole todistuva/johdettavissa.

3. **Semanttinen taulu**
   + Lauseissa voi esiintyä kaikkia peruskonnektiiveja.
   + Lauseiden syntaksi ohjaa päättelysäännön (taulusäännöt) valintaa.
   + Mikäli lause ei ole todistuva/johdettavissa saadaan konkreettinen vastaesimerkki (eli vastamalli $\mathcal{A}$).
   + Helppo toteuttaa tietokoneella.
   – Semanttinen taulu ei ole tehokkain mahdollinen menetelmä lauselogiikan päättelyongelmien ratkomiseen.

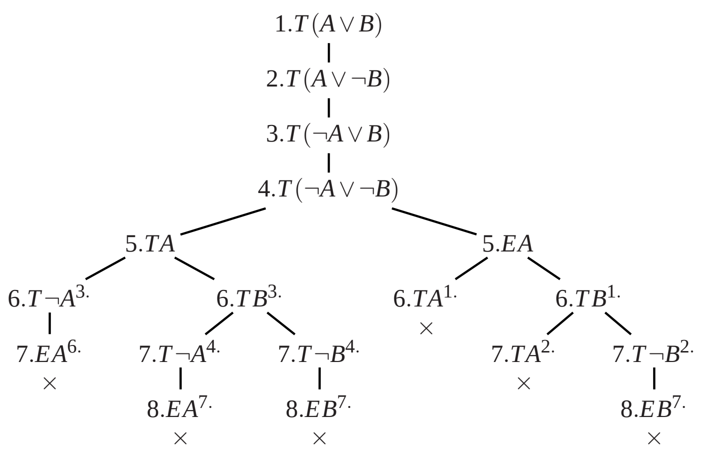
Huomio. Semanttisen taulun menetelmää voidaan edelleen tehostaa lisäämällä taulusääntöjä. Esimerkkinä mainittakoon ns. *cut-sääntö*, joka haarauttaa polun jonkin väittämän α suhteen ($T\alpha$ ja $E\alpha$).

Harkiten käytettynä tällä säännöllä voi tiivistää todistuksia.

Esimerkki. Pahimmassa tapauksessa taulun koko voi kasvaa eksponentiaalisesti atomisten lauseiden lukumäärään nähden.

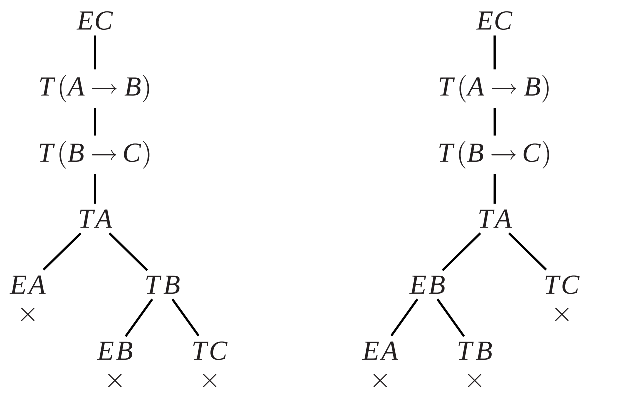


Esimerkki. Cut-säännön avulla taulu jää hieman pienemmäksi:



Merkitys korostuisi, jos atomisten lauseitten määrää lisättäisiin.

Esimerkki. Semanttisella taululla pystytään *simuloimaan* monia Suppes-järjestelmän päättelysääntöjä.



"Modus ponens" "Modus tollendo tollens"

6 Normaalimuodot

- Konjunktiivinen ja disjunktiivinen normaalimuoto
- Muunnokset normaalimuotoihin
- Normaalimuotojen hakeminen taulumenetelmällä
- Normaalimuotojen sieventäminen
- Lauseiden klausuulimuoto

Motivaatio

Normaalimuotojen tarkoituksena on saattaa käsiteltävät lausekkeet johonkin säännölliseen muotoon, jotta niiden käsittely yksinkertaistuu. Esimerkkeinä mainittakoon seuraavat.

- Polynomien esittäminen muodossa

$$c_n \cdot x^n + c_{n-1} \cdot x^{n-1} + \cdots + c_0 \cdot x^0.$$

- Lineaaristen yhtälöryhmien esittäminen matriiseina:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

6.1 Konjunktiivinen ja disjunktiivinen normaalimuoto

Määritelmä. Jos A on atominen lause, niin A ja $\neg A$ ovat *literaaleja*.

Määritelmä. *Positiivisen* literaalin A *komplementti* $\overline{A} = \neg A$ ja *negatiivisen* literaalin $\neg A$ *komplementti* $\overline{\neg A} = A$.

Määritelmä. Lause α on *konjunktiivisessa normaalimuodossa*, jos α on muodoltaan konjunktio $\beta_1 \wedge \beta_2 \wedge \cdots \wedge \beta_n$, missä jokainen β_i on literaaleista $l_1, l_2, \ldots, l_{m_i}$ koostuva disjunktio $l_1 \vee l_2 \vee \cdots \vee l_{m_i}$.

Disjunktiivinen normaalimuoto määritellään samaan tapaan, mutta konjunktion ja disjunktion roolit vaihdetaan keskenään:

Määritelmä. Lause α on *disjunktiivisessa normaalimuodossa*, jos α on muodoltaan disjunktio $\beta_1 \vee \beta_2 \vee \cdots \vee \beta_n$, missä jokainen β_i on literaaleista $l_1, l_2, \ldots, l_{m_i}$ koostuva konjunktio $l_1 \wedge l_2 \wedge \cdots \wedge l_{m_i}$.

Esimerkki.

KNM:	$(A_1 \lor \neg A_2) \land (\neg A_1 \lor \neg A_3) \land (A_5 \lor A_6 \lor A_7)$
	$A_5 \land (\neg A_3 \lor A_6)$
KNM & DNM:	$A_1 \lor \neg A_2 \lor A_3$
	$\neg A_7$
	$A_1 \land A_2 \land A_3$
DNM:	$(\neg A_1 \land \neg A_2) \lor (A_3 \land A_1) \lor (A_1 \land \neg A_2 \land A_3)$
	$(A_1 \land \neg A_1) \lor A_2$

Väite. Jokaiselle lauselogiikan lauseelle α on olemassa konjunktiivisessa (disjunktiivisessa) normaalimuodossa oleva lause β, joka on loogisesti ekvivalentti α:n kanssa.

Määritelmä. Sanotaan, että em. β on α:n konjunktiivinen (disjunktiivinen) normaalimuoto.

Esimerkki. $A \lor (\neg B \land C)$ on loogisesti ekvivalentti konjunktiivisessa normaalimuodossa olevan lauseen $(A \lor \neg B) \land (A \lor C)$ kanssa.

A	B	C	$\neg B$	$\neg B \land C$	$A \lor \neg B$	$A \lor C$	$A \lor (\neg B \land C)$	$(A \lor \neg B) \land (A \lor C)$
E	E	E	T	E	T	E	E	E
E	E	T	T	T	T	T	T	T
E	T	E	E	E	E	E	E	E
E	T	T	E	E	E	T	E	E
T	E	E	T	E	T	T	T	T
T	E	T	T	T	T	T	T	T
T	T	E	E	E	T	T	T	T
T	T	T	E	E	T	T	T	T

6.2 Muunnokset normaalimuotoihin

Mikä hyvänsä lauselogiikan lause voidaan muuttaa konjunktiiviseen tai disjunktiiviseen normaalimuotoon seuraavalla menettelyllä.

1. Poista konnektiivit $\leftrightarrow$ seuraavanlaisilla korvauksilla:

   $\alpha \leftrightarrow \beta \quad \rightsquigarrow \quad (\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha) \qquad (1)$

   $\alpha \leftrightarrow \beta \quad \rightsquigarrow \quad (\neg\alpha \lor \beta) \land (\neg\beta \lor \alpha) \qquad (1')$

2. Poista konnektiivit $\rightarrow$ seuraavanlaisilla korvauksilla:

   $\alpha \rightarrow \beta \quad \rightsquigarrow \quad \neg\alpha \lor \beta \qquad (2)$

3. Siirrä negaatiot välittömästi atomisten lauseiden eteen:

   $\neg\neg\alpha \quad \rightsquigarrow \quad \alpha \qquad (3)$

   $\neg(\alpha \lor \beta) \quad \rightsquigarrow \quad \neg\alpha \land \neg\beta \qquad (4)$

   $\neg(\alpha \land \beta) \quad \rightsquigarrow \quad \neg\alpha \lor \neg\beta \qquad (5)$

Viimeinen vaihe riippuu tavoiteltavasta normaalimuodosta:

4. (KNM) Siirrä $\land$-konnektiivit $\lor$-konnektiivien ulkopuolelle:

   $\alpha \lor (\beta \land \gamma) \quad \rightsquigarrow \quad (\alpha \lor \beta) \land (\alpha \lor \gamma) \qquad (6)$

   $(\alpha \land \beta) \lor \gamma \quad \rightsquigarrow \quad (\alpha \lor \gamma) \land (\beta \lor \gamma) \qquad (7)$

   (DNM) Siirrä $\lor$-konnektiivit $\land$-konnektiivien ulkopuolelle:

   $\alpha \land (\beta \lor \gamma) \quad \rightsquigarrow \quad (\alpha \land \beta) \lor (\alpha \land \gamma) \qquad (8)$

   $(\alpha \lor \beta) \land \gamma \quad \rightsquigarrow \quad (\alpha \land \gamma) \lor (\beta \land \gamma) \qquad (9)$

Huomio. Jokainen edellä annetuista muunnoksista säilyttää loogisen ekvivalenssin, joten lopputuloksena saadaan lauseelle konjunktiivinen tai disjunktiivinen normaalimuoto.

Esimerkki. Muutetaan lause $A \vee B \rightarrow (B \leftrightarrow C)$ konjunktiiviseen ja disjunktiiviseen normaalimuotoon.

$$A \vee B \rightarrow (B \rightarrow C) \wedge (C \rightarrow B) \quad (1)$$
$$\neg(A \vee B) \vee ((\neg B \vee C) \wedge (\neg C \vee B)) \quad (2)$$
$$(\neg A \wedge \neg B) \vee ((\neg B \vee C) \wedge (\neg C \vee B)) \quad (4)$$

Konjunktiivinen normaalimuoto:

$$(\neg A \vee ((\neg B \vee C) \wedge (\neg C \vee B))) \wedge (\neg B \vee ((\neg B \vee C) \wedge (\neg C \vee B))) \quad (7)$$
$$((\neg A \vee (\neg B \vee C)) \wedge (\neg A \vee (\neg C \vee B))) \wedge (\neg B \vee ((\neg B \vee C) \wedge (\neg C \vee B))) \quad (6)$$
$$((\neg A \vee (\neg B \vee C)) \wedge (\neg A \vee (\neg C \vee B))) \wedge ((\neg B \vee (\neg B \vee C)) \wedge (\neg B \vee (\neg C \vee B))) \quad (6)$$
$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg C \vee B) \wedge (\neg B \vee \neg B \vee C) \wedge (\neg B \vee \neg C \vee B)$$

Disjunktiivinen normaalimuoto:

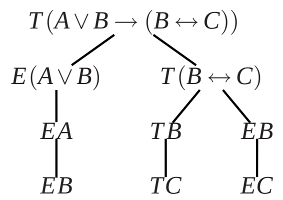
$$(\neg A \wedge \neg B) \vee ((\neg B \vee C) \wedge \neg C) \vee ((\neg B \vee C) \wedge B) \quad (8)$$
$$(\neg A \wedge \neg B) \vee (\neg B \wedge \neg C) \vee (C \wedge \neg C) \vee (\neg B \wedge B) \vee (C \wedge B) \quad (9)$$

6.3 Normaalimuotojen hakeminen taulumenetelmällä

Normaalimuodon voi hakea myös taulumenetelmällä.

Lauseen α disjunktiivinen normaalimuoto saadaan juurisolmusta $T\alpha$ muodostetun valmiin semanttisen taulun ristiriidattomista poluista.

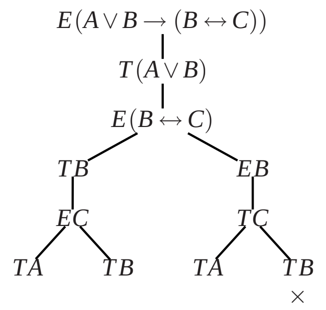
Esimerkki.



Vasemmanpuolisella polulla on vaatimukset EA ja EB, keskimmäisellä TB ja TC ja oikeanpuolisella EB ja EC. Näistä saadaan lauseen disjunktiiviseksi normaalimuodoksi $(\neg A \wedge \neg B) \vee (B \wedge C) \vee (\neg B \wedge \neg C)$.

Lauseelle α saadaan konjunktiivinen normaalimuoto lauseen $\neg\alpha$ disjunktiivisesta normaalimuodosta sääntöjen (3)-(5) avulla.

Esimerkki.



Ristiriidattomista poluista saadaan DNM lauseelle $\neg\alpha$:
$\neg\alpha \equiv (A \wedge B \wedge \neg C) \vee (B \wedge \neg C) \vee (A \wedge \neg B \wedge C)$.
Lauseen α konjunktiivinen normaalimuoto on tällöin:
$\alpha \equiv \neg\neg\alpha \equiv (\neg A \vee \neg B \vee C) \wedge (\neg B \vee C) \wedge (\neg A \vee B \vee \neg C)$.

6.4 Normaalimuotojen sieventäminen

Konjunktiivista/disjunktiivista normaalimuotoa voidaan sieventää mm. seuraavilla periaatteilla:

- Poistetaan literaalien disjunktioista/konjunktioista literaalien moninkertaiset esiintymät.

  Esimerkki.
  $(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg C \vee B) \wedge (\neg B \vee \neg B \vee C) \wedge (\neg B \vee \neg C \vee B)$
  $\rightsquigarrow\ (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg C \vee B) \wedge (\neg B \vee C) \wedge (\neg B \vee \neg C \vee B)$

- Poistetaan literaalien disjunktiot/konjunktiot, joissa esiintyy jokin literaali l ja sen komplementti $\bar{l}$.

  Esimerkki. Jatketaan edellisestä:
  $\rightsquigarrow\ (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg C \vee B) \wedge (\neg B \vee C)$

- Jos **konjunktiivisessa** normaalimuodossa esiintyy disjunktiot $(l_1 \vee \cdots \vee l_n)$ ja $(k_1 \vee \cdots \vee k_m)$ s.e. $\{l_1,\ldots,l_n\} \subseteq \{k_1,\ldots,k_m\}$, poistetaan jälkimmäinen (joka on edellisen looginen seuraus).

  Esimerkki. $(A \vee B \vee C) \wedge (A \vee B) \wedge (C \vee \neg B \vee A) \wedge (A \vee C)$
  $\rightsquigarrow (A \vee B) \wedge (C \vee \neg B \vee A) \wedge (A \vee C) \rightsquigarrow (A \vee B) \wedge (A \vee C)$

- Jos **konjunktiivisessa** normaalimuodossa esiintyy yksiliteraalinen disjunktio l, poistetaan muut disjunktiot, joissa l esiintyy, sekä mahdolliset komplementin $\bar{l}$ esiintymät muista disjunktioista.

  Esimerkki. $\neg A \wedge (A \vee B \vee C) \wedge (\neg A \vee \neg B) \rightsquigarrow \neg A \wedge (B \vee C)$

  Huomio. Jos disjunktiosta poistetaan kaikki jäsenet, jäljelle jää tyhjä disjunktio $\perp$, joka on aina epätosi.

  Esimerkki. $A \wedge B \wedge (\neg A \vee \neg B) \rightsquigarrow A \wedge B \wedge \neg B \rightsquigarrow A \wedge B \wedge \perp \rightsquigarrow \perp$ (alkuperäinen lause on siis toteutumaton).

6.5 Lauseiden klausuulimuoto

- Atomiset lauseet A ja niiden negaatiot $\neg A$ ovat *literaaleja*.
- Literaalin komplementti: $\overline{A} = \neg A$, $\overline{\neg A} = A$.
- Literaalien $l_1,\ldots,l_n$ disjunktio $l_1 \vee \cdots \vee l_n$ on *klausuuli*.
- Klausuulit esitetään usein literaalien *joukkoina* $\{l_1,\ldots,l_n\}$.
- Joukko klausuuleita S edustaa klausuuliensa konjunktiota.

Huomio. Esitystavoilla on hienoiset erot:

$$\neg A \vee B \vee \neg A \rightsquigarrow \{\neg A, B\}$$
$$\neg A \vee B \rightsquigarrow \{\neg A, B\}$$
$$B \vee \neg A \rightsquigarrow \{\neg A, B\}$$

Konjunktiivisesta normaalimuodosta voidaan muodostaa vastaava klausuulijoukko.

Esimerkki. Konjunktiivista normaalimuotoa $(A \vee \neg B) \wedge (\neg C \vee \neg A \vee D)$ vastaava klausuulijoukko on $\{\{A, \neg B\}, \{\neg A, \neg C, D\}\}$

Huomaa seuraavat erikoistapaukset:

- *Tyhjä klausuuli* $\square$ edustaa tyhjää disjunktiota (ja on siten aina epätosi).
- Tyhjä klausuulijoukko $\emptyset$ edustaa tyhjää konjunktiota (ja on siten aina tosi).

Huomio. Muistanet seuraavan analogian matematiikasta (tyhjät summat ja tulot): $\sum_{i=1}^{n} x_i = 0$ ja $\prod_{i=1}^{n} x_i = 1$, kun $n = 0$.

7 Resoluutio

- Totuusmääritelmä ja toteutuvuus klausuuleille
- Resoluutiosääntö
- Resoluutiotodistukset
- Resoluution virheettömyys ja täydellisyys
- Loogisten ongelmien ratkominen resoluutiolla

7.1 Totuusmääritelmä ja toteutuvuus klausuuleille

Klausuulien tapauksessa totuusmääritelmä voidaan kirjoittaa varsin yksinkertaiseen muotoon.

Määritelmä.

1. Totuusjakelun $\mathcal{A} \subseteq \mathcal{P}$ *literaaliesitys* $\mathrm{Lit}(\mathcal{A}) = \mathcal{A} \cup \{\neg A \mid A \in \mathcal{P} - \mathcal{A}\}$.
2. Klausuuli C on *tosi* totuusjakelussa $\mathcal{A} \subseteq \mathcal{P}$ (merk. $\mathcal{A} \models C$), joss $C \cap \mathrm{Lit}(\mathcal{A}) \neq \emptyset$.
3. Klausuulijoukko S on *tosi* totuusjakelussa $\mathcal{A}$ (merk. $\mathcal{A} \models S$), joss kaikille klausuuleille $C \in S$ pätee $\mathcal{A} \models C$.
4. Totuusjakelu $\mathcal{A} \subseteq \mathcal{P}$ on klausuulijoukon S *malli*, joss $\mathcal{A} \models S$.

Esimerkki. Tarkastellaan atomisten lauseiden joukkoon $\mathcal{P} = \{A, B, C, D, E\}$ perustuvia klausuuleita.

Olkoon $\mathcal{A} = \{A\}$, jolloin $\mathrm{Lit}(\mathcal{A}) = \{A, \neg B, \neg C, \neg D, \neg E\}$.

Nyt esimerkiksi

$$\begin{array}{lcl}
\mathcal{A} & \models & \{\{A, C, E\}, \{\neg A, \neg B\}\} \\
\mathcal{A} & \models & \emptyset \\
\mathcal{A} & \not\models & \{\{A, B\}, \{C\}\} \\
\mathcal{A} & \models & \{\neg A, B, \neg D\} \\
\mathcal{A} & \not\models & \{\neg A, D\} \\
\mathcal{A} & \not\models & \square
\end{array}$$

Määritelmä. Klausuulijoukko S on *toteutuva* joss S:llä on ainakin yksi malli $\mathcal{A} \subseteq \mathcal{P}$. Muuten S on *toteutumaton*.

Esimerkki. Klausuulijoukot $\{\{A, C, E\}, \{\neg A, \neg B\}\}$ ja $\emptyset$ ovat toteutuvia, koska esimerkiksi $\mathcal{A} = \{A\}$ on näiden molempien malli.

Esimerkki. Klausuulijoukot $\{\{A\}, \{\neg A\}\}$ ja $\{\square\}$ ovat toteutumattomia, koska molemmista klausuulijoukoista löytyy klausuulit C_1 ja C_2 siten, että $C_1 \cap \{A\} = \emptyset$ ja $C_2 \cap \{\neg A\} = \emptyset$.

Huomaa, että edellä $\{A\} = \mathrm{Lit}(\{A\})$ ja $\{\neg A\} = \mathrm{Lit}(\emptyset)$ kattavat kaikki mahdolliset totuusjakelut $\mathcal{A} \subseteq \mathcal{P} = \{A\}$.

Esimerkki. Aiemmin esitetyn kartan kolmiväritysongelman kuvaus vastaa seuraavaa toteutumatonta klausuulijoukkoa $S =$

$$\begin{array}{l}
\{\ \{P_1, V_1, S_1\},\ \{\neg P_1, \neg V_1\},\ \{\neg V_1, \neg S_1\},\ \{\neg S_1, \neg P_1\}, \\
\quad \{P_2, V_2, S_2\},\ \{\neg P_2, \neg V_2\},\ \{\neg V_2, \neg S_2\},\ \{\neg S_2, \neg P_2\}, \\
\quad \{P_3, V_3, S_3\},\ \{\neg P_3, \neg V_3\},\ \{\neg V_3, \neg S_3\},\ \{\neg S_3, \neg P_3\}, \\
\quad \{P_4, V_4, S_4\},\ \{\neg P_4, \neg V_4\},\ \{\neg V_4, \neg S_4\},\ \{\neg S_4, \neg P_4\}, \\
\quad \{\neg P_1, \neg P_2\},\ \{\neg V_1, \neg V_2\},\ \{\neg S_1, \neg S_2\}, \\
\quad \{\neg P_1, \neg P_3\},\ \{\neg V_1, \neg V_3\},\ \{\neg S_1, \neg S_3\}, \\
\quad \{\neg P_1, \neg P_4\},\ \{\neg V_1, \neg V_4\},\ \{\neg S_1, \neg S_4\}, \\
\quad \{\neg P_2, \neg P_3\},\ \{\neg V_2, \neg V_3\},\ \{\neg S_2, \neg S_3\}, \\
\quad \{\neg P_2, \neg P_4\},\ \{\neg V_2, \neg V_4\},\ \{\neg S_2, \neg S_4\}, \\
\quad \{\neg P_3, \neg P_4\},\ \{\neg V_3, \neg V_4\},\ \{\neg S_3, \neg S_4\}\ \}.
\end{array}$$

7.2 Resoluutiosääntö

Määritelmä. Olkoot $C_1 = \{l, l_1, \ldots, l_n\}$ ja $C_2 = \{\bar{l}, l'_1, \ldots, l'_m\}$ klausuuleja. Klausuuli $C = (C_1 \cup C_2) - \{l, \bar{l}\} = \{l_1, \ldots, l_n, l'_1, \ldots, l'_m\}$ on klausuulien C_1 ja C_2 *yhdistelmä*.

Esimerkki. Sovelletaan resoluutiosääntöä seuraaviin klausuuleihin:

$$\begin{array}{ccc} \{A, B, \neg C\} & & \{\neg A, D, \neg C\} \\ & \searrow \quad \swarrow & \\ & \{B, \neg C, D\} & \end{array}$$

Sääntöä on sovellettu literaalien A ja $\neg A$ suhteen. Klausuulit ovat joukkoja, joten $\neg C$ esiintyy yhdistelmässä $\{B, \neg C, D\}$ vain kertaalleen.

Huomio. Resoluutiosääntöä saa soveltaa korkeintaan yhden literaaliparin (l ja $\bar{l}$) suhteen kerrallaan.

Esimerkki. Tarkastellaan klausuulijoukkoa $S = \{\{A, \neg B\}, \{\neg A, B\}\}$.

- Klausuuljoukosta voidaan johtaa resoluutiosäännöllä klausuulit $\{A, \neg A\}$ (literaali $l = B$) ja $\{B, \neg B\}$ (literaali $l = A$).
- Edelleen näistä ja joukon S klausuuleista voidaan johtaa resoluutiosäännöllä ainoastaan S:ään kuuluvia klausuuleita.
- Missään tapauksessa S:stä ei saada resoluutiosäännöllä tyhjää klausuulia $\Box$ (tämä tarkoittaisi, että S on toteutumaton).
- Huomaa, että S on toteutuva ($\mathcal{A} \models S$) totuusjakelussa $\mathcal{A} = \{A, B\}$.

7.3 Resoluutiotodistukset

Lähtökohtana klausuulijoukko S, jonka klausuuleihin sovelletaan resoluutiosääntöä.

Määritelmä. Klausuulin C *johto* klausuulijoukosta S on äärellinen jono klausuuleja $C_1, \ldots, C_n$, missä $C_n = C$ ja jokaiselle C_i joko $C_i \in S$ tai C_i on joidenkin aikaisempien klausuulien yhdistelmä.

Määritelmä. Jos klausuulijoukosta S voidaan johtaa tyhjä klausuuli $\Box$, kyseistä johtoa kutsutaan S:n *hylkäykseksi* (refutaatioksi).

Ajatuksena on, että tällöin S joudutaan hylkäämään toteutuvana klausuulijoukkona.

Esimerkki. Tarkastellaan klausuulijoukkoa

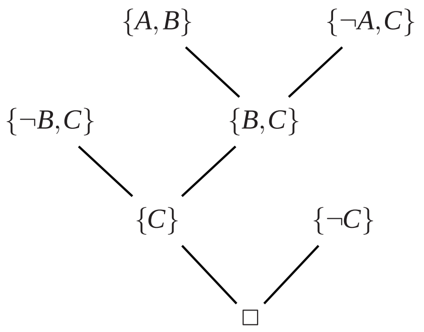
$$S = \{\{A, B\}, \{\neg A, C\}, \{\neg B, C\}, \{\neg C\}\},$$

jolle saadaan seuraava hylkäys:

1.	$\{A, B\}$	S
2.	$\{\neg A, C\}$	S
3.	$\{\neg B, C\}$	S
4.	$\{\neg C\}$	S
5.	$\{B, C\}$	1, 2
6.	$\{C\}$	3, 5
7.	$\Box$	4, 6

Resoluutiotodistuksen puuesitys

Edellä esitetty hylkäys voidaan kirjoittaa myös puumuotoon.



Huomio. Lehtisolmuina on ainoastaan klausuulijoukon S klausuleja ja vastaava lineaarinen hylkäys voidaan tuottaa käymällä puumuotoinen todistus syvyysjärjestyksessä lävitse.

Esimerkki. Klausuulijoukolle

$$S = \{ \{A,B,\neg C,\neg D\}, \{\neg A,E\}, \{\neg B,\neg F\}, \{C,E\}, \{D,\neg F\}, \{\neg E\}, \{F\} \}$$

saadaan seuraava hylkäys:

1.	$\{A,B,\neg C,\neg D\}$	S
2.	$\{\neg A,E\}$	S
3.	$\{E,B,\neg C,\neg D\}$	1,2
4.	$\{\neg B,\neg F\}$	S
5.	$\{E,\neg F,\neg C,\neg D\}$	3,4
6.	$\{C,E\}$	S
7.	$\{E,\neg F,\neg D\}$	5,6
8.	$\{D,\neg F\}$	S
9.	$\{E,\neg F\}$	7,8
10.	$\{\neg E\}$	S
11.	$\{\neg F\}$	9,10
12.	$\{F\}$	S
13.	$\square$	11, 12

7.4 Resoluution virheettömyys ja täydellisyys

Väite. Jos klausuulijoukolla S on malli $\mathcal{A} \subseteq \mathcal{P}$ ja C on kahden klausuulin $C_1 \in S$ ja $C_2 \in S$ yhdistelmä, niin totuusjakelu $\mathcal{A}$ on myös laajennetun klausuulijoukon $S' = S \cup \{C\}$ malli.

Todistus. Oletetaan $\mathcal{A} \not\models S \cup \{C\}$.

$\Longrightarrow \quad \mathcal{A} \not\models C$

$\Longrightarrow \quad C \cap \mathrm{Lit}(\mathcal{A}) = \emptyset$

$\Longrightarrow \quad C_1 \cap \mathrm{Lit}(\mathcal{A}) = \emptyset$ **tai** $C_2 \cap \mathrm{Lit}(\mathcal{A}) = \emptyset$

$\Longrightarrow \quad \mathcal{A} \not\models C_1$ tai $\mathcal{A} \not\models C_2$

$\Longrightarrow \quad \mathcal{A} \not\models S$, ristiriita.

Siis $\mathcal{A} \models S \cup \{C\}$.

Väite. Jos klausuulijoukolle S on hylkäys, niin S on toteutumaton.

Todistus. Oletetaan, että klausuulijoukolle S on hylkäys $C_1, \cdots, C_n$, missä $C_n = \square$. Tehdään vastaoletus, että S on toteutuva.

Osoitetaan induktiolla i:n suhteen, että $S \cup \{C_1, \ldots, C_i\}$ on toteutuva.

Perustapaus $i = 0$: Joukko S on toteutuva (vastaoletus).

Induktioaskel: Joukko $S \cup \{C_1, \ldots, C_{i-1}\}$ on toteutuva (induktio-oletus).

1. Jos $C_i \in S$, joukko $S \cup \{C_1, \ldots, C_i\}$ on triviaalisti toteutuva.
2. Muussa tapauksessa C_i on saatu resoluutiosäännöllä joukon $\{C_1, \ldots, C_{i-1}\}$ kahdesta klausuulista. Edellä todistamamme väitteen nojalla myös joukko $S \cup \{C_1, \ldots, C_i\}$ on toteutuva.

Induktiotodistuksesta seuraa, että myös $S \cup \{C_1, \ldots, C_n\}$ on toteutuva.

Ristiriita, koska $C_n = \square$ on epätosi kaikissa totuusjakeluissa.

Puukonstruktio täydellisyystodistusta varten

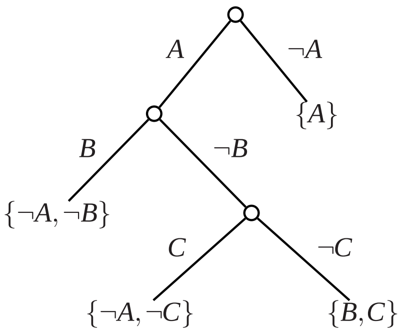
Olkoon S atomisten lauseiden joukkoon $\mathcal{P} = \{A_1, \ldots, A_n\}$ perustuva klausuulijoukko, jolle muodostetaan binääripuu seuraavilla periaatteilla.

Olkoon s syvyydellä i $(0 \leq i \leq n)$ oleva puun solmu (juurisolmu on syvyydellä 0) ja L_s niiden literaalien joukko, jotka ovat juurisolmusta solmuun s johtavilla kaarilla.

Jos $\overline{C} = \{\overline{l} \mid l \in C\} \subseteq L_s$ jollekin klausuulille $C \in S$ (eli $\mathcal{A} \not\models C$ kaikille totuusjakeluille $\mathcal{A}$ s.e. $L_s \subseteq \mathrm{Lit}(\mathcal{A})$), merkitään C solmuun s ja lopetetaan puun laajentaminen tästä solmusta eteenpäin. Muutoin:

1. Jos $i < n$, merkitään solmulle s vasen lapsi s_v ja oikea lapsi s_o sekä merkitään näihin solmusta s johtaville kaarille literaalit A_i ja $\neg A_i$. Jatketaan puun laajentamista solmuista s_v ja s_o vastaavasti.
2. Jos $i = n$, lopetetaan puun laajentaminen solmusta s eteenpäin.

Esimerkki. Konstruoidaan edellä kuvattu binääripuu klausuulijoukolle $S = \{\{A\}, \{B, C\}, \{\neg A, \neg B\}, \{\neg A, \neg C\}\}$:



Huomio. Kuhunkin lehtisolmuun s merkitty klausuuli C on epätosi totuusjakeluissa $\mathcal{A}$, joille $L_s \subseteq \mathrm{Lit}(\mathcal{A})$.

Väite. Jos S on toteutumaton, niin binääripuun jokainen polku päättyy solmuun s, johon on merkitty klausuuli $C \in S$ s.e. $\overline{C} \subseteq L_s$.

Todistus. Vastaoletus: edellä kuvatulla tavalla muodostetussa binääripuussa on solmu s tasolla n siten, että $\overline{C} \not\subseteq L_s$ kaikille $C \in S$.
$\Longrightarrow$ $L_s = \mathrm{Lit}(\mathcal{A})$ jollekin $\mathcal{A} \subseteq \mathcal{P}$ ja $C \cap L_s \neq \emptyset$ kaikille $C \in S$.
$\Longrightarrow$ $\mathcal{A} \models C$ kaikille $C \in S$, eli S on toteutuva, ristiriita.

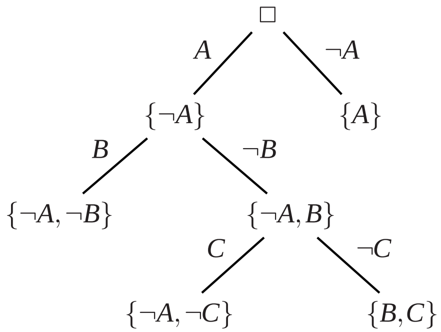
Väite. Jos klausuulijoukko S on toteutumaton, sille on hylkäys.

Todistus. Olkoon S toteutumaton, jolloin binääripuun lehtisolmuina on S:n klausuulit. Käydään läpi sisäsolmut s (käänteisessä järjestyksessä).

Merkitään solmuun s klausuuliksi lapsisolmuihin merkittyjen klausuulien C_v ja C_o yhdistelmä C, jolle pätee $\overline{C} \subseteq L_s$ ($\overline{C_v} \subseteq L_{s_v}$ ja $\overline{C_o} \subseteq L_{s_o}$). Tämä ominaisuus siirtyy kaikille sisäsolmujen s klausuuleille C.

Juurisolmun s tapauksessa $L_s = \emptyset$, joten $\overline{C} = \emptyset$ ja $C = \Box$.

Esimerkki. Palataan edelliseen esimerkkiin ja muodostetaan ko. klausuulijoukolle $S = \{\{A\}, \{B, C\}, \{\neg A, \neg B\}, \{\neg A, \neg C\}\}$ hylkäys:



Huomio. Tästä on helppo todeta edellä esitetty ominaisuus, että jokaiseen solmuun s merkitylle klausuulille C pätee $\overline{C} \subseteq L_s$.

7.5 Loogisten ongelmien ratkominen resoluutiolla

Toteutuvuuden tutkiminen resoluutiolla

- Menettely: muunnetaan tutkittava lause α konjunktiiviseen normaalimuotoon ja edelleen klausuulijoukoksi S_α.
- Tällöin S_α on toteutumaton
  $\iff$ lauseen α KNM on toteutumaton
  $\iff$ α on toteutumaton.
- Sen sijaan mallien hakeminen resoluutiolla on hankalaa.

Lausejoukkojen toteutuvuutta tutkitaan samaan tyyliin.

- Menettely: muodostetaan klausuulijoukko $S_\Sigma = \bigcup\{S_\sigma \mid \sigma \in \Sigma\}$.
- Tällöin S_Σ on toteutumaton $\iff$ Σ on toteutumaton.

Esimerkki. Osoitetaan karttaesimerkin klausuulijoukko toteutumattomaksi resoluutiolla.

1. $\{P_4, V_4, S_4\}$
2. $\{\neg S_3, \neg S_4\}$
3. $\{P_4, V_4, \neg S_3\}$ (1,2)
4. $\{P_3, V_3, S_3\}$
5. $\{P_3, V_3, P_4, V_4\}$ (3,4)
6. $\{\neg V_2, \neg V_4\}$
7. $\{P_3, V_3, P_4, \neg V_2\}$ (5,6)
8. $\{\neg V_2, \neg V_3\}$
9. $\{P_3, P_4, \neg V_2\}$ (7,8)
10. $\{\neg V_3, \neg V_4\}$
11. $\{P_4, S_4, \neg V_3\}$ (1,10)
12. $\{P_3, S_3, P_4, S_4\}$ (4,11)
13. $\{\neg S_2, \neg S_4\}$
14. $\{P_3, S_3, P_4, \neg S_2\}$ (12,13)
15. $\{\neg S_2, \neg S_3\}$
16. $\{P_3, P_4, \neg S_2\}$ (14,15)
17. $\{P_2, V_2, S_2\}$
18. $\{P_2, P_3, P_4, V_2\}$ (16,17)

19. $\{P_2, P_3, P_4\}$ (9,18)
20. $\{\neg P_1, \neg P_2\}$
21. $\{P_3, P_4, \neg P_1\}$ (19,20)
22. $\{\neg P_1, \neg P_3\}$
23. $\{P_4, \neg P_1\}$ (21,22)
24. $\{\neg P_1, \neg P_4\}$
25. $\{\neg P_1\}$ (23,24)

...

...

50. $\{\neg V_1\}$ (symmetria)

...

75. $\{\neg S_1\}$ (symmetria)
76. $\{P_1, V_1, S_1\}$
77. $\{V_1, S_1\}$ (25,76)
78. $\{S_1\}$ (50,77)
79. $\square$ (75,78)

- Edellä olevan todistuksen hakemisessa (askelet 1–25) on hyödynnetty täydellisyystodistuksen puukonstruktiota.
- Lyhyempiäkin todistuksia löydettävissä (OTTER: 49 askelta).

Pätevyyden tutkiminen resoluutiolla

Väite. $\models \alpha \iff \neg\alpha$ on toteutumaton.

- Menettely: muunnetaan tutkittavan lauseen negaatio $\neg\alpha$ konjunktiiviseen normaalimuotoon ja edelleen klausuulijoukoksi $S_{\neg\alpha}$.
- Tällöin $S_{\neg\alpha}$ on toteutumaton
  $\iff$ lauseen $\neg\alpha$ KNM on toteutumaton
  $\iff$ $\neg\alpha$ on toteutumaton
  $\iff$ α on pätevä.

Esimerkki. Onko $(A \to B) \wedge (\neg A \to C) \to B \vee C$ pätevä?

Lauseen negaation KNM on $(\neg A \vee B) \wedge (A \vee C) \wedge \neg B \wedge \neg C$.

Klausuulijoukko $S = \{\{\neg A, B\}, \{A, C\}, \{\neg B\}, \{\neg C\}\}$.

Resoluutiotodistus (hylkäys):

1.	$\{\neg A, B\}$	S
2.	$\{A, C\}$	S
3.	$\{\neg B\}$	S
4.	$\{\neg C\}$	S
5.	$\{B, C\}$	1, 2
6.	$\{C\}$	3, 5
7.	$\square$	4, 6

Lause on siis pätevä.

Esimerkki. Onko $(A \to B) \wedge (A \to C) \to B \vee C$ pätevä?

Lauseen negaation KNM on $(\neg A \vee B) \wedge (\neg A \vee C) \wedge \neg B \wedge \neg C$.

Resoluutiotodistus:

1.	$\{\neg A, B\}$	S
2.	$\{\neg A, C\}$	S
3.	$\{\neg B\}$	S
4.	$\{\neg C\}$	S
5.	$\{\neg A\}$	1, 3

Muita klausuuleja (mukaanlukien $\square$) ei ole enää johdettavissa.

Lause ei ole pätevä.

Vastamalli $\mathcal{A} = \emptyset$ nähtävissä klausuuleista 3–5.

Loogisen seuraavuuden tutkiminen resoluutiolla

Väite. $\Sigma \models \alpha \iff \Sigma \cup \{\neg\alpha\}$ on toteutumaton.

- Menettely: muutetaan lausejoukon $\Sigma \cup \{\neg\alpha\}$ lauseet konjunktiiviseen normaalimuotoon ja edelleen klausulijoukoksi $S_{\Sigma \cup \{\neg\alpha\}}$.
- Tällöin $S_{\Sigma \cup \{\neg\alpha\}}$ on toteutumaton
  $\iff$ joukon $\Sigma \cup \{\neg\alpha\}$ lauseiden KMN:en joukko on toteutumaton
  $\iff$ lausejoukko $\Sigma \cup \{\neg\alpha\}$ on toteutumaton
  $\iff$ α on lausejoukon Σ looginen seuraus.

Esimerkki. Onko $\{\neg A \to B, B \vee C \to \neg B\} \models A$?

lause	KNM	klausuuleina
$\neg A$	$\neg A$	$\{\neg A\}$
$\neg A \to B$	$A \vee B$	$\{A, B\}$
$B \vee C \to \neg B$	$(\neg B \vee \neg B) \wedge (\neg C \vee \neg B)$	$\{\neg B\}, \{\neg B, \neg C\}$

Saadaan klausuulijoukko $S = \{\{\neg A\}, \{A, B\}, \{\neg B\}, \{\neg B, \neg C\}\}$.

1.	$\{\neg A\}$	S
2.	$\{A, B\}$	S
3.	$\{\neg B, \neg C\}$	S
4.	$\{\neg B\}$	S
5.	$\{B\}$	1, 2
6.	$\square$	4, 5

Vastaus: lause A on lausejoukon looginen seuraus.

Esimerkki. Palataan hissiesimerkin spesifikaatioon ja osoitetaan ko. turvallisuusominaisuus resoluutiolla:

lause	KNM	klausuuleina	
$\neg K_1 \vee \neg K_2$	$\neg K_1 \vee \neg K_2$	1. $\{\neg K_1, \neg K_2\}$	
$A_1 \rightarrow K_1$	$\neg A_1 \vee K_1$	2. $\{\neg A_1, K_1\}$	
$A_2 \rightarrow K_2$	$\neg A_2 \vee K_2$	3. $\{\neg A_2, K_2\}$	
$\neg\neg(A_1 \wedge A_2)$	$A_1 \wedge A_2$	4. $\{A_1\}$, 5. $\{A_2\}$	
	Hylkäys:	6. $\{K_1\}$	2,4
		7. $\{K_2\}$	3,5
		8. $\{\neg K_2\}$	1,6
		9. $\square$	7,8

$\Longrightarrow$ $\neg(A_1 \wedge A_2)$ on muiden lauseiden looginen seuraus.

8 Laskennallisesta vaativuudesta

- Laskennan malli
- Keskeiset vaativuusluokat
- Redusoituvuus ja vaikeat ongelmat

8.1 Laskennan malli

Oletamme jatkossa, että laskennan mallina ovat *Turing-koneet*.

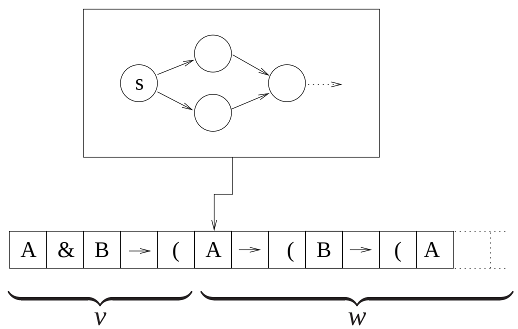
Määritelmä.
Deterministinen Turing-kone T on nelikkö $\langle A, S, s_0, t\rangle$, missä

- A on *aakkosto*, johon kuuluu aina erikoissymboli $\sqcup$ (tyhjä symboli).
- S on joukko tiloja, johon kuuluu aina annettu *alkutila* $s_0 \in S$ sekä erikoistilat k (kyllä), e (ei) ja p (pysähdy).
- $t : S \times A \rightarrow S \times A \times \{\leftarrow, \rightarrow, \downarrow\}$ on *tilansiirtofunktio*.

Huomio. Tyhjää merkkijonoa merkitään symbolilla ε.

Määritelmä. Turing-koneen T kokonaistilan määrää *konfiguraatio* $\langle s, v, w\rangle$, missä $s \in S$ on T:n tila ja $v \in A^*$ ja $w \in A^+$ ovat merkkijonoja.

Merkkijonot v ja w ovat peräkkäin koneen T työnauhalla:



T käsittelee aina merkkijonon w ensimmäistä merkkiä, jonka kohdalla koneen luku-kirjoituspään ajatellaan sijaitsevan.

Laskennan määritelmä

- Laskenta alkaa konfiguraatiosta $\langle s_0, \varepsilon, w \rangle$, missä merkkijono $w \in (A - \{\sqcup\})^*$ tai $w = \sqcup$ (T:n syöte).
- Yhdessä laskennan askeleessa siirrytään konfiguraatiosta $\langle s, v, aw \rangle$ tilansiirtofunktion t arvon $t(s,a) = \langle s', a', m \rangle$ perusteella uuteen konfiguraatioon seuraavasti:
  1. Jos $m = \downarrow$, uusi konfiguraatio on $\langle s', v, a'w \rangle$.
  2. Jos $m = \rightarrow$, uusi konfiguraatio on $\langle s', va', w' \rangle$, missä $w' = w$, jos $w \neq \varepsilon$, ja $w' = \sqcup$, jos $w = \varepsilon$.
  3. Jos $m = \leftarrow$ ja $v = v'b$ joillekin $v' \in A^*$ ja $b \in A$, uusi konfiguraatio on $\langle s', v', ba'w \rangle$.
- Laskenta päättyy, jos $s' \in \{k, e, p\}$.

Määritelmä. Turing-koneen T laskenta on sekvenssi konfiguraatioita $\langle s_0, v_0, w_0 \rangle \xrightarrow{T} \ldots \xrightarrow{T} \langle s_{n-1}, v_{n-1}, w_{n-1} \rangle$ missä $v_0 = \varepsilon$ ja $s_{n-1} \in \{k, e, p\}$. Kone T hyväksyy syötteen w_0, joss $s_{n-1} = k$.

Esimerkki. Olkoon $A = \{0, 1, \sqcup\}$ ja $S = \{s_0, s_1, k, e, p\}$. Binääriluvun pariteetti voidaan tarkastaa seuraavalla Turing-koneella T_{par}:

S	A	$S \times A \times \{\rightarrow, \leftarrow, \downarrow\}$
s_0	0	$\langle s_0, 0, \rightarrow \rangle$
s_0	1	$\langle s_1, 1, \rightarrow \rangle$
s_0	$\sqcup$	$\langle k, \sqcup, \downarrow \rangle$
s_1	0	$\langle s_1, 0, \rightarrow \rangle$
s_1	1	$\langle s_0, 1, \rightarrow \rangle$
s_1	$\sqcup$	$\langle e, \sqcup, \downarrow \rangle$

Syötteellä 101 T_{par} suorittaa seuraavan laskennan:
$\langle s_0, \varepsilon, 101 \rangle$
$\xrightarrow{T_{\text{par}}} \langle s_1, 1, 01 \rangle$
$\xrightarrow{T_{\text{par}}} \langle s_1, 10, 1 \rangle$
$\xrightarrow{T_{\text{par}}} \langle s_0, 101, \sqcup \rangle$
$\xrightarrow{T_{\text{par}}} \langle k, 101, \sqcup \rangle$.

Epädeterministiset Turing-koneet

- Tilansiirtofunktio t korvataan tilansiirtorelaatiolla $t : S \times A \to 2^{S \times A \times \{\leftarrow, \rightarrow, \downarrow\}}$.
- Konfiguraatiossa $\langle s, v, aw \rangle$ valitaan epädeterministisesti $\langle s', a', m \rangle \in t(s,a)$ ja siirrytään tämän perusteella uuteen konfiguraatioon. Mahdollisia laskentoja voi olla useita.

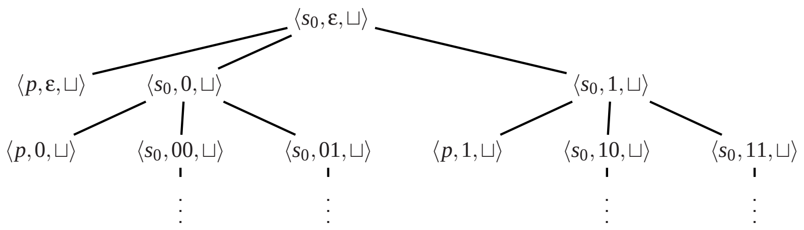
Esimerkki. Olkoon $A = \{0, 1, \sqcup\}$ ja $S = \{s_0, k, e, p\}$. Määritellään epädeterministinen Turing-kone seuraavasti:

S	A	$2^{S \times A \times \{\leftarrow, \rightarrow, \downarrow\}}$
s_0	$\sqcup$	$\{\langle s_0, 0, \rightarrow \rangle, \langle s_0, 1, \rightarrow \rangle, \langle p, \sqcup, \downarrow \rangle\}$

Yksi mahdollinen laskenta: $\langle s_0, \varepsilon, \sqcup \rangle \xrightarrow{T} \langle s_0, 1, \sqcup \rangle$
$\xrightarrow{T} \langle s_0, 10, \sqcup \rangle \xrightarrow{T} \langle s_0, 101, \sqcup \rangle \xrightarrow{T} \langle p, 101, \sqcup \rangle$.

Epädeterministisen laskennan piirteitä

- Vaihtoehtoisista laskennoista muodostuu *laskentapuu*:



Huomio. Deterministisellä Turing-koneella on vain yksi mahdollinen laskenta, joten vastaavassa laskentapuussa on vain yksi haara.

Kielen hyväksyminen Turing-koneella

- Deterministinen Turing-kone T hyväksyy syötteen $x \in (A - \{\sqcup\})^*$, joss koneen T laskenta syötteellä x pysähtyy tilaan k.
- Deterministinen Turing-kone T *hyväksyy kielen* $L \subseteq (A - \{\sqcup\})^*$, joss kaikille $x \in (A - \{\sqcup\})^*$: kone T hyväksyy syötteen $x \iff x \in L$.

  Esimerkki. Edellä esitetty binääriluvun pariteetin tarkastava kone T_{par} hyväksyy vastaavan kielen $L_{\text{parillinen}} \subseteq \{0, 1\}^*$.
- Epädeterministisellä koneella hyväksyviä laskentoja voi olla useita ja täten hyväksymisehdosta tulee monimutkaisempi.

Määritelmä. (Epädeterministinen) Turing-kone T *hyväksyy* kielen $L \subseteq (A - \{\sqcup\})^*$, jos kaikille merkkijonoille $x \in (A - \{\sqcup\})^*$ pätee: koneella T on ainakin yksi hyväksyvä laskenta syötteellä $x \iff x \in L$.

Turing-koneiden käyttö päätösongelmien ratkaisemiseen

- *Päätösongelmat* ovat ongelmia, joiden instanssien ratkaisuksi riittää yksinkertainen vastaus “kyllä” tai “ei”.

  Esimerkki. Onko 561 alkuluku?
- Päätösongelman O ratkaiseminen Turing-koneella edellyttää ongelmainstanssien esittämistä merkkijonoina ja Turing-koneen T konstruointia “kyllä”-instansseja vastaavan kielen hyväksymiseen.

Esimerkki. Lauselogiikan toteutuvuusongelmassa SAT:ssa on tarkoituksena selvittää, onko annettu lause $\phi \in \mathcal{L}$ toteutuva vai ei.

SAT-ongelmaa vastaava kieli (“kyllä”-instanssien joukko) on siis toteutuvien lauseiden ϕ joukko (lauseet merkkijonoesityksinä).

$\Longrightarrow$ SAT-ongelma voidaan samaistaa em. kielen kanssa.

8.2 Keskeiset vaativuusluokat

- Päätösongelmien laskennallista vaativuutta voidaan analysoida asettamalla Turing-koneiden laskentaresursseille rajoituksia.

  Määritelmä. Turing-kone T pysähtyy polynomisessa ajassa syötteen pituuden suhteen, joss on olemassa polynomi p siten, että kaikilla syötteillä $w \in (A - \{\sqcup\})^*$ koneen T laskenta käsittää korkeintaan $p(|w|)$ erilaista konfiguraatiota.
- Kaksi keskeistä ongelmien luokkaa ovat

  $\mathbf{P}$: päätösongelmat, joiden instanssit voidaan ratkaista polynomisessa ajassa *deterministisellä* Turing-koneella.

  $\mathbf{NP}$: päätösongelmat, joiden instanssit voidaan ratkaista polynomisessa ajassa *epädeterministisellä* Turing-konella.
- Luokka $\mathbf{P}$ on luokan $\mathbf{NP}$ aliluokka (ja mitä ilmeisimmin aito).

Väite. SAT kuuluu luokkaan $\mathbf{NP}$.

Todistuksen idea:

Voidaan konstruoida epädeterministinen Turing-kone T, joka

- valitsee epädeterministisesti totuusjakelun $\mathcal{A}$,
- laskee syötteenä ϕ annetun lauseen totuusarvon $\mathcal{A}$:ssa ja
- pysähtyy tilaan k, jos $\mathcal{A} \models \phi$, ja muutoin tilaan e.

Tarvittava laskenta pystytään suorittamaan polynomisessa ajassa lauseen ϕ merkkijonoesityksen pituuden suhteen.

Voidaan osoittaa, että $\phi \in$ SAT $\iff$ koneella T on ainakin yksi hyväksyvä laskenta syötteellä ϕ.

8.3 Redusoituvuus ja vaikeat ongelmat

- Ongelman O_1 *redusoituvuus* ongelmaksi O_2 edellyttää, että löytyy instanssin i pituuden suhteen polynomisessa ajassa deterministisellä Turing-koneella laskettava funktio $r : O_1 \to O_2$ siten, että
$$i \in O_1 \iff r(i) \in O_2.$$
- Tällaista funktiota r kutsutaan usein *reduktioksi*.

Esimerkki. Olkoon $G = \langle S, K \rangle$ graafi, missä $K \subseteq S \times S$ antaa kaaret.

Ongelma graafin G kolmiväritettävyydestä (3COL) voidaan redusoida lauselogiikan toteutuvuusongelmaksi seuraavasti: $r(G) = r(\langle S, K \rangle) =$

$$\{M_s \vee V_s \vee H_s \mid s \in S\} \cup \{\neg M_s \vee \neg M_t,\ \neg V_s \vee \neg V_t,\ \neg H_s \vee \neg H_t \mid \langle s, t \rangle \in K\}.$$

Nyt voidaan osoittaa, että $G \in \text{3COL} \iff r(G) \in \text{SAT}$.

Vaikeat ja täydelliset ongelmat

Määritelmä. Olkoon $\mathbf{C}$ jokin luokka ongelmia (kuten $\mathbf{P}$ tai $\mathbf{NP}$). Ongelma O on

1. $\mathbf{C}$-*vaikea*, joss jokainen luokan $\mathbf{C}$ ongelma O' voidaan redusoida O:ksi polynomisessa ajassa ja
2. $\mathbf{C}$-*täydellinen*, joss O kuuluu luokkaan $\mathbf{C}$ ja O on $\mathbf{C}$-vaikea.

$\Longrightarrow$ $\mathbf{C}$-täydelliset ongelmat ovat vaativimpia luokan $\mathbf{C}$ ongelmia.

Väite. SAT on $\mathbf{NP}$-vaikea (Cook, 1971).

Todistuksen idea: Jokaiselle epädeterministiselle Turing-koneelle T, merkkijonolle $w \in (A - \{\sqcup\})^*$ ja polynomille p löytyy lausejoukko Σ s.e.

- koneella T on syötteellä w ainakin yksi hyväksyvä laskenta, jonka pituus on pienempi kuin $p(|w|) \iff$ lausejoukko Σ on toteutuva.

$\Longrightarrow$ SAT on $\mathbf{NP}$-täydellinen ongelma.

Huomioita.

- Näin ollen *epädeterministisen* Turing-koneen suorittama polynominen, syötteen hyväksyvä laskenta voidaan palauttaa polynomisessa ajassa lauselogiikan toteutuvuusongelman ratkaisemiseen.
- Nykykäsitysten mukaisesti SAT-ongelman ratkaiseminen *deterministisellä* Turing-koneella vaatii pahimmassa tapauksessa eksponentiaalisen ajan lauseen pituuteen nähden.
- Tehokas algoritmi: Davis-Putnam-Logemann-Loveland [1961,1962]
- Käytettävällä hakuheuristiikalla on kuitenkin olennainen vaikutus DPLL-algoritmin suorituskykyyn.

DPLL-algoritmi

Olkoon S äärellinen klausuulijoukko ja L joukko literaaleja.

Funktio OnToteutuva(S, L): { K, E };
Aloita
  $\langle S, L \rangle$:= Yksinkertaista(S, L);
  Jos $\Box \in S$, **Niin Vastaa** E;
  Jos $S = \emptyset$, **Niin Vastaa** K;
  l := Valitse(S, L);
  Jos OnToteutuva(S, $L \cup \{l\}$), **Niin Vastaa** K
  Muutoin Vastaa OnToteutuva(S, $L \cup \{\bar{l}\}$);
Lopeta

Huomioita

- Funktion Yksinkertaista(S, L) tehtävänä on yksinkertaistaa klausuulijoukkoa S ja laajentaa literaalijoukkoa L hyödyntämällä literaaleja l, joille $l \in L$ tai klausuuli $\{l\} \in S$, rekursiivisesti:
  1. Lisätään l joukkoon L, mikäli se ei siellä vielä ole.
  2. Poistetaan joukosta S toteutuneet klausuulit $C \in S$, joille $l \in C$, ja literaalin $\bar{l}$ esiintymät S:n klausuuleista (ovat epätosia).
- Funktio Valitse(S, L) toteuttaa *hakuheuristiikan* eli valitsee literaalin l (s.e. $l \notin L$ ja $\bar{l} \notin L$), joka esiintyy joukossa S ja jonka suhteen haku seuraavaksi haarautuu.
- Tehokkaita DPLL-toteutuksia, näiden vertailuja ja benchmark-ongelmia on saatavilla verkossa; kts. esim. `www.satlive.org` ja `www.satcompetition.org`.