Lauselogiikan ratkaisumenetelmiä

➤ Resoluutio:

$$\frac{A \vee L_1^1 \vee \cdots \vee L_n^1, \neg A \vee L_1^2 \vee \cdots \vee L_m^2}{L_1^1 \vee \cdots \vee L_n^1 \vee L_1^2 \vee \cdots \vee L_m^2}$$

➤ Semanttiset taulut

➤ Binääriset päätösdiagrammit (BDD)

➤ DP(DLL)-menetelmä

➤ Paikallinen haku

➤ Satunnaistetut menetelmät

➤ Optimointimenetelmät

☞ Täydellisiä menetelmiä

☞ Epätäydellisiä menetelmiä

DP-menetelmä

➤ Tehokkaimmat nykyiset lauselogiikan ratkaisumenetelmät pohjautuvat DP-menetelmään [Davis-Logemann-Loveland, 1962], jota usein (hieman epätarkasti) kutsutaan Davis-Putnam-menetelmäksi [1960].

➤ Refutaatio-menetelmä: lause P todistetaan aloittamalla lauseesta $\neg P$ ja johtamalla ristiriita:

Lause $\neg P$ muunnetaan konjunktiiviseen normaalimuotoon klausuulijoukoksi, joka osoitetaan toteutumattomaksi.

➤ Menetelmällä voidaan myös hakea klausuulijoukolle mallia/malleja.

➤ Täydellinen menetelmä: päättää jokaisen klausuulijoukon osalta, onko joukko toteutuva vai ei.

Perusidea

➤ Klausuulijoukon toteutuvuutta tutkitaan muodostamalla *DP-taulu*: puu, jonka juurisolmuna on tutkittava klausuulijoukko.

➤ Taulun solmun (klausuulijoukon) lapset saadaan muuntamalla ko. solmua muunnossäännöillä.

➤ Muunnokset säilyttävät toteutuvuuden:
Jos solmu (klausuulijoukko) on toteutuva, ainakin yksi sen muunnoksella saaduista lapsista on myös toteutuva.

➤ Klausuulijoukkoja muunnetaan, kunnes on ilmeistä, ovatko ne toteutuvia vai ei (eliminoimalla toteutuneet klausuulit ja muista klausuuleista epätodet literaalit):
(i) Tyhjä klausuulijoukko on toteutuva.
(ii) Tyhjän klausuulin sisältävä klausuulijoukko ei ole toteutuva.

Määritelmiä

➤ Literaalin L *komplementti* $\overline{L}$:
Olkoon A atomilause. $\overline{A} = \neg A$ ja $\overline{\neg A} = A$.

➤ *Tyhjä klausuuli*: $\perp$
Esim. Jos klausuulista $A \vee \neg B$ poistetaan literaalit A ja $\neg B$, jäljelle jää tyhjä klausuuli $\perp$.

Tyhjä klausuuli $\perp$ on jokaisessa mallissa epätosi.

➤ Esikäsittely:

(i) Poistetaan kustakin klausuulista moneen kertaan esiintyvät literaalit.

(ii) Poistetaan *tautologiset* (aina todet) klausuulit, joissa esiintyy literaali ja sen komplementti.

Muunnossääntöjä

- **Yhden literaalin sääntö (One-literal rule):**
  (Yksikköliteraalisääntö)

  Olkoon klausuulijoukossa S yhden literaalin klausuuli L.

  Muunnetaan S poistamalla
  (i) kaikki klausuulit, jotka sisältävät literaalin L ja
  (ii) muista klausuuleista literaali $\overline{L}$ (yksikköresoluutio).

- *Malliehto:* L tosi.

Esimerkki. Olkoon $S = \{\neg A, A \vee B \vee C, \neg A \vee D\}$.

Yhden literaalin sääntö muuntaa joukon S muotoon $\{B \vee C\}$.

Malliehto: $\neg A$ tosi.

- **Komplementti puuttuu -sääntö (Affirmative-negative rule):**

  Olkoon klausuulijoukossa S klausuuli, joka sisältää literaalin L muttei yhtään klausuulia, joka sisältää literaalin $\overline{L}$.

  Muunnetaan S poistamalla kaikki klausuulit, jotka sisältävät literaalin L.

- *Malliehto:* L tosi.

Esimerkki. Olkoon $S = \{A \vee \neg B \vee C, \neg A \vee \neg C\}$.

Komplementti puuttuu -sääntö muuntaa joukon S muotoon $\{\neg A \vee \neg C\}$.

Malliehto: $\neg B$ tosi.

- **Peittosääntö (Subsumption rule):**

  Olkoon klausuulijoukossa S klausuulit C_1 ja C_2 siten, että C_2 *peittää* klausuulin C_1 (jokainen klausuulin C_1 literaali esiintyy myös klausuulissa C_2).

  Muunnetaan S poistamalla C_2.

- *Ei malliehtoa!*

Esimerkki. Olkoon $S = \{A \vee \neg B \vee \neg C, A \vee \neg C, \neg A \vee \neg B\}$.

Peittosääntö muuntaa joukon S muotoon

$\{A \vee \neg C, \neg A \vee \neg B\}$.

Ei malliehtoa.

- **Haarautumissääntö (Splitting rule):**

  Olkoon klausuulijoukossa S klausuuli, joka sisältää literaalin L ja klausuuli, joka sisältää literaalin $\overline{L}$.

  Muunnetaan S kahdeksi joukoksi S_L ja $S_{\overline{L}}$.

- S_L saadaan joukosta S poistamalla kaikki klausuulit, jotka sisältävät literaalin L ja muista klausuuleista literaali $\overline{L}$
  Malliehto: L tosi.

- $S_{\overline{L}}$ saadaan joukosta S poistamalla kaikki klausuulit, jotka sisältävät literaalin $\overline{L}$ ja muista klausuuleista literaali L
  Malliehto: $\overline{L}$ tosi.

Esimerkki. Joukosta $S = \{A \vee \neg B \vee \neg C, \ \neg A \vee \neg B\}$ saadaan:

$S_A = \{\neg B\}$ (malliehto: A tosi) ja

$S_{\neg A} = \{\neg B \vee \neg C\}$ (malliehto: $\neg A$ tosi).

DP-taulut

Määritelmä. Klausuulijoukoista muodostuva binääripuu T on DP-taulu, jos

1. T:n ainoana solmuna (eli sekä juuri- että lehtisolmuna) on mikä tahansa klausuulijoukko S, tai
2. T' on DP-taulu ja T saadaan soveltamalla johonkin T':n lehtisolmuna olevaan klausuulijoukkoon S muunnossääntöä ja liittämällä syntyvät 1 tai 2 klausuulijoukkoa solmun S lapsiksi.

➤ DP-taulun haara, jonka lehtisolmu on S,
(i) on onnistunut, jos $\perp \in S$ ja (ii) epäonnistunut, jos $S = \emptyset$.
Taulu on onnistunut, jos kaikki sen haarat ovat onnistuneita.

➤ Klausuulijoukko S on toteutumaton joss on olemassa onnistunut DP-taulu T, jonka juurena on S. (Tulos pätee vaikka käytössä olisi vain haarautumissääntö).

DP-todistukset

DP-todistus lauseelle P saadaan muuntamalla $\neg P$ klausuulijoukoksi S ja etsimällä onnistunut DP-taulu, jonka juurena on S.
Teoreema. Lause P on pätevä joss sille on olemassa DP-todistus.

Esimerkki. Todistetaan $\phi = ((P \rightarrow Q) \wedge (Q \rightarrow P) \wedge (\neg P \rightarrow Q)) \rightarrow (P \wedge Q)$.

(i) $\neg\phi$ klausuulimuotoon:

$$\rightsquigarrow \quad (P \rightarrow Q) \wedge (Q \rightarrow P) \wedge (\neg P \rightarrow Q) \wedge \neg(P \wedge Q)$$
$$\rightsquigarrow \quad \{\neg P \vee Q,\ \neg Q \vee P,\ P \vee Q,\ \neg P \vee \neg Q\}.$$

Muodostetaan onnistunut DP-taulu: $\{\neg P \vee Q,\ \neg Q \vee P,\ P \vee Q,\ \neg P \vee \neg Q\}$
(malliehdot annettu suluissa)

(P)	$(\neg P)$
$\{Q, \neg Q\}$	$\{\neg Q, Q\}$
(Q)	$(\neg Q)$
$\{\perp\}$	$\{\perp\}$

DP-taulujen ominaisuuksia

➤ Epäonnistunut DP-johto antaa klausuulijoukolle mallin:
Otetaan malli, joka täyttää tyhjään klausuulijoukkoon johtaneen muunnossääntöketjun yhteydessä saadut malliehdot.

➤ DP-tauluja voidaan käyttää myös mallin etsimiseen klausuulijoukolle S:
Joukko S taulun juureksi ja etsitään epäonnistunutta haaraa (jonka lehtisolmu on tyhjä).

➤ Vaikka DP-tauluissa käytettävät muunnossäännöt säilyttävät toteutuvuuden, klausuulijoukkojen mallit eivät välttämättä säily.

➤ Erityisesti komplementti puuttuu -sääntö sulkee pois malleja.

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

1. Soveltamalla komplementti puuttuu -sääntöä A:n suhteen, saadaan klausuulijoukoksi $S' = \emptyset$ ja malliehdoksi A.
2. Koska lopputuloksena on tyhjä klausuulijoukko, malliehdon A nojalla klausuulijoukolle S voidaan muodostaa kaksi mallia $M_1 = \{A\}$ ja $M_2 = \{A, B\}$.
3. Klausuulijoukolla on kuitenkin kolmas malli $M_3 = \{B\}$.

Esimerkki. Tutkitaan, onko lause $\phi =$

$$((R \rightarrow P \vee Q) \wedge (P \vee Q) \wedge (R \rightarrow \neg Q) \wedge (Q \vee S) \wedge (S \rightarrow R)) \rightarrow (S \wedge R)$$

pätevä.

1. Haetaan $\neg\phi$:lle klausuulimuoto: $S =$

$$\{\neg R \vee P \vee Q,\ P \vee Q,\ \neg R \vee \neg Q,\ Q \vee S,\ \neg S \vee R,\ \neg S \vee \neg R\}$$

2. Muodostetaan DP-taulu (seuraava kalvo).
3. Epäonnistunut haara (klausuulijoukko tyhjä) antaa alkuperäiselle klausuulijoukolle mallin $\mathcal{A} = \{P, Q\}$.
4. Koska klausuulijoukko S on ekvivalentti lauseen $\neg\phi$ kanssa, alkuperäiselle lauseelle ϕ saatiin näin vastamalli $\mathcal{A}$.

➤ DP-taulusta muodostuu seuraava:

$$\begin{array}{c}
\{\neg R \vee P \vee Q,\ P \vee Q,\ \neg R \vee \neg Q,\ Q \vee S,\ \neg S \vee R,\ \neg S \vee \neg R\} \\
\text{(peittosääntö)} \\
\{P \vee Q,\ \neg R \vee \neg Q,\ Q \vee S,\ \neg S \vee R,\ \neg S \vee \neg R\} \\
(P) \\
\{\neg R \vee \neg Q,\ Q \vee S,\ \neg S \vee R,\ \neg S \vee \neg R\} \\
\begin{array}{c|c}
(\neg R) & (R) \\
\{Q \vee S,\ \neg S\} & \{\neg Q,\ Q \vee S, \neg S\} \\
(\neg S) & (\neg Q) \\
\{Q\} & \{S, \neg S\} \\
(Q) & (S) \\
\{\} & \{\bot\}
\end{array}
\end{array}$$

DP-taulut vs. semanttiset taulut

➤ Semanttisissa tauluissa ei yleensä käytetä haarautumissääntöä.

➤ DP-tauluilla voidaan simuloida polynomisesti klausuulijoukolle muodostettavaa semanttista taulua. Esimerkiksi

$$\begin{array}{ccc}
\begin{array}{cc}
\multicolumn{2}{c}{T(L_1 \vee C_2)} \\
/ & \backslash \\
TL_1 & TC_2
\end{array}
& \mapsto &
\begin{array}{c|c}
\multicolumn{2}{c}{\{\ldots, L_1 \vee C_2, \ldots\}} \\
(L_1) & (\overline{L_1}) \\
\{\ldots, \ldots\} & \{\ldots, C_2, \ldots\}
\end{array}
\end{array}$$

➤ Semanttisella taululla ei voi simuloida polynomisesti DP-taulua. Esimerkki: $\{\neg P \vee Q,\ \neg Q \vee P,\ P \vee Q,\ \neg P \vee \neg Q\}$

☞ DP-menetelmä on vahvempi todistusmenetelmä kuin taulut ilman haarautumissääntöä.

Muita sääntöjä

➤ **Literaali epäonnistuu -sääntö:**
Jos tyhjä klausuuli saadaan joukosta $S \cup \{L\}$ käyttämällä yhden literaalin sääntöä, poistetaan kaikki klausuulit, joissa esiintyy $\overline{L}$ ja muista klausuuleista literaali L.

➤ *Malliehto:* $\overline{L}$.

Esimerkki. Olkoon $S = \{A \vee B,\ \neg B \vee C,\ \neg C \vee A\}$.

Yhden literaalin säännöllä:

$S \cup \{\neg A\} \rightsquigarrow \{B,\ \neg B \vee C,\ \neg C\} \rightsquigarrow \{C,\ \neg C\} \rightsquigarrow \{\bot\}$

Saadaan muunnettu klausuulijoukko $S = \{\neg B \vee C\}$.

Malliehto: A.

- **Literaalipari epäonnistuu -sääntö:**

  Jos klausuulijoukosta $S \cup \{L_1, L_2\}$ saadaan käyttämällä yhden literaalin sääntöä $\perp$, lisätään joukkoon S klausuuli $\overline{L_1} \vee \overline{L_2}$.

- **Poista 1-literaalit -sääntö:**

  Literaali L esiintyy täsmälleen yhdessä S:n klausuulissa $L \vee C_0$ ja $\overline{L}$ esiintyy k klausuulissa $\overline{L} \vee C_1, \ldots, \overline{L} \vee C_k$.

  Korvataan joukossa S em. $k+1$ klausuulia k klausuulilla $C_0 \vee C_1, \ldots, C_0 \vee C_k$.

Esimerkki. Olkoon $S = \{A \vee \neg B \vee C, \neg C \vee A\}$.
Yhden literaalin säännöllä:

$$S \cup \{\neg A, B\} \rightsquigarrow \{B, \neg B \vee C, \neg C\} \rightsquigarrow \{C, \neg C\} \rightsquigarrow \{\perp\}$$

Joten joukkoon S lisätään klausuuli $A \vee \neg B$.

DP-menetelmän toteuttaminen

- Yksinkertaisuutensa vuoksi DP-menetelmä on toteutettavissa varsin suoraviivaisesti.
- Ylärajana hakuavaruuden koolle on 2^v (v on atomilauseiden lkm.).
- Käyttämällä syyvyyshakua algoritmi voidaan toteuttaa polynomisessa tilassa klausuulijoukon kokoon nähden.
- Jokaisessa hakuavaruuden pisteessä:
  (a) yksinkertaistetaan klausuulijoukkoa ja
  (b) valitaan uusi atomilause haarautumista varten.
- Hakuheuristiikka vaikuttaa suorituskykyyn olennaisesti.
- Pahimmassa tapauksessa algoritmin suoritus vaatii eksponentiaalisen ajan klausuulijoukon kokoon nähden.

Perusalgoritmi

Syöte: klausuulijoukko S ja literaalijoukko M (malliehdot).
function $\mathrm{DP}(S, M)$: literaalijoukko tai arvo 'UNSAT';
begin
  $\langle S', M' \rangle := \mathrm{simplify}(S, M)$;
  if $S' = \emptyset$ **then** return M'
  else if $\perp \in S'$ **then** return 'UNSAT'
  else begin
    $L := \mathrm{choose}(S', M')$;
    $M'' := \mathrm{DP}(S' \cup \{L\}, M' \cup \{L\})$;
    if M'' = 'UNSAT' **then** return $\mathrm{DP}(S' \cup \{\overline{L}\}, M' \cup \{\overline{L}\})$
    else return M''
  end
end

- Funktio $\mathrm{simplify}(S, M)$ pyrkii
  1. yksinkertaistamaan klausuulijoukkoa S ja
  2. täydentämään malliehtojen joukkoa M.
- Funktio $\mathrm{choose}(S, M)$ valitsee
  1. atomilauseen A haarautumissääntöä varten ja
  2. ensimmäisenä käsiteltävän malliehdon (A tai $\neg A$).
- Suorituskyky riippuu olennaisesti heuristiikasta.
- Kaupankäynti: yksinkertaistaminen vs. haarautuminen.
  - Yksinkertaistamisella voidaan vähentää haarautumistarvetta ja siten pienentää käsiteltävää hakuavaruutta.
  - Esim. yhden literaalin sääntöä kannattaa käyttää jokaisessa hakuavaruuden pisteessä.

Esimerkki. Olkoon $S = \{A, \neg A \vee B, \neg B \vee C, \neg C\}$.

- Kyseinen klausuulijoukko voidaan osoittaa toteutumattomaksi yhden literaalin säännöllä (malliehdot: A, B ja C).

$$S \rightsquigarrow \{B, \neg B \vee C, \neg C\} \rightsquigarrow \{C, \neg C\} \rightsquigarrow \{\bot\}$$

- Toteutumattomuus voidaan osoittaa myös haarautumissäännöllä:

$$\begin{array}{c}
\{A, \neg A \vee B, \neg B \vee C, \neg C\}\\
\begin{array}{c|c}
\begin{array}{c}(A)\\ \{B, \neg B \vee C, \neg C\}\\ \begin{array}{c|c} \begin{array}{c}(B)\\ \{C, \neg C\}\\ \begin{array}{c|c}(C) & (\neg C)\\ \{\bot\} & \{\bot\}\end{array}\end{array} & \begin{array}{c}(\neg B)\\ \{\bot, \neg C\}\\ \\ \\ \end{array}\end{array}\end{array} & \begin{array}{c}(\neg A)\\ \{\bot, \neg B \vee C, \neg C\}\\ \\ \\ \\ \\ \end{array}
\end{array}
\end{array}$$

Tyypillinen toteutus

1. Tehdään esikäsittely, jossa yksinkertaistetaan klausuulijoukkoa ja vähennetään muuttujien määrää (polynominen aikavaativuus).

   Esim. käyttämällä yhden literaalin, komplementti puuttuu, literaali epäonnistuu ja poista 1-literaalit sääntöä.

2. Jokaisessa hakupisteessä yksinkertaistetaan klausuulijoukkoa: tyypillisesti ainakin yhden literaalin sääntö (toteutettava tehokkaasti).

3. Kussakin hakupisteessä lasketaan heuristiikka (literaalien esiintymistiheyksiä klausuuleissa ja/tai mitä johtopäätöksiä saataisiin yhden literaalin säännöllä).

4. Hakua voidaan edelleen tehostaa
   – käyttämällä oikaisevaa palautuvaa hakua (backjumping) ja
   – lisäämällä klausuuleja (rajoitettu oppiminen).