Davis-Putnam -menetelmä

- Tehokkaimmat nykyiset lauselogiikan ratkaisumenetelmät pohjautuvat Davis-Putnam menetelmään [1960] (oikeastaan [Davis-Logemann-Loveland, 1962]).
- 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

- Tutkitaan klausuulijoukkojen joukon toteutuvuutta:

  Klausuulijoukkojen joukko on toteutuva joss ainakin yksin sen klausuulijoukoista on 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.
- Muunnokset säilyttävät toteutuvuuden:

  Jos klausuulijoukko on ennen muunnosta toteutuva, on se myös sen jälkeen.

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.

Muunnossäännöt

Yhden literaalin sääntö (One-literal rule):

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.

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

Peittosääntö muuntaa joukon S muotoon

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

Huom! 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 $S_{\overline{L}}$ missä

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

Haarautumissääntö muuntaa joukon S kahdeksi joukoksi

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

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

DP-johdot

- *Rintama* R on joukko klausuulijoukkoja S ja R ymmärretään klausuulijoukkojensa *disjunktiona*.
- Klausuulijoukosta S alkava *DP-johto* on jono $R_1, R_2, R_3, \ldots$ rintamia, missä $R_1 = \{S\}$ ja kun $i > 0$, rintama R_i on saatu rintamasta R_{i-1} korvaamalla jokin klausuulijoukoista $S' \in R_{i-1}$ niillä 1:llä tai 2:lla klausuulijoukolla, jotka saadaan soveltamalla joukkoon S' yhtä edellä kuvatuista säännöistä.

DP-johtojen ominaisuuksia

Rintama R on

- *valmis*, jos sen jokaiselle klausuulijoukolle $S \in R$ pätee *joko* $\bot \in S$ *tai* $S = \emptyset$.
- *onnistunut*, jos sen jokaiselle klausuulijoukolle $S \in R$ pätee $\bot \in S$.
- *epäonnistunut*, jos siihen sisältyy tyhjä klausuulijoukko $S = \emptyset$.

Huomaa, että jos R on

- onnistunut, R on klausuulijoukkojensa disjunktiona toteutumaton.
- epäonnistunut, R on klausuulijoukkojensa disjunktiona toteutuva.

DP-johto $R_1, R_2, R_3, \ldots$ on valmis/onnistunut/epäonnistunut, jos se päättyy rintamaan R, joka on valmis/onnistunut/epäonnistunut.

Esimerkki (onnistunut DP-johto).

$$\begin{aligned} R_1 &= [\{P \vee Q, P \vee \neg Q, \neg P \vee Q, \neg P \vee \neg Q\}] \\ R_2 &= [\{Q, \neg Q\}, \{Q, \neg Q\}] \\ R_3 &= [\{\bot\}, \{Q, \neg Q\}] \\ R_4 &= [\{\bot\}, \{\bot\}] \end{aligned}$$

Esimerkki (epäonnistunut DP-johto).

$$\begin{aligned} R_1 &= [\{P \vee Q, P \vee \neg Q, \neg P \vee Q\}] \\ R_2 &= [\{Q\}, \{Q, \neg Q\}] \\ R_3 &= [\{Q\}, \{\bot\}] \\ R_4 &= [\emptyset, \{\bot\}] \end{aligned}$$

Väittämä 1 (DP-johdon ominaisuudet I)
Olkoon S äärellinen klausuulijoukko ja $R_1, R_2, R_3, \ldots,$ rintamasta $R_1 = [S]$ lähtien muodostettu DP-johto. Jos kyseinen DP-johto $R_1, R_2, R_3, \ldots,$ on valmis, niin se on myös (i) äärellinen ja (ii) joko onnistunut tai epäonnistunut.

Todistus. (i) Sääntöjä voidaan soveltaa vain äärellisen monta kertaa, koska S on äärellinen ja jokainen muunnossääntö vähentää atomilauseiden määrää tai klausuulien määrää (peittosääntö). Näin DP-johto muodostuu äärelliseksi jonoksi rintamia $R_1, \ldots, R_n$.

(ii) Oletetaan, että johto $R_1, \ldots, R_n$ ei ole onnistunut eikä epäonnistunut. Tällöin on olemassa ei-tyhjä klausuulijoukko $S \in R_n$, joka ei sisällä tyhjää klausuulia. Johto $R_1, \ldots, R_n$ ei siis ole valmis. Johtoa voitaisiin jatkaa valitsemalla klausuuli $C \in S$ ja C:n literaali L. Jos literaali $\overline{L}$ esiintyy S:ssä, voidaan käyttää haarautumissääntöä, muutoin komplementti puuttuu -sääntöä.

Väittämä 2 (DP-johdon ominaisuudet II)
Olkoon S äärellinen klausuulijoukko ja $R_1, \ldots, R_n$ rintamasta $R_1 = [S]$ modostettu valmis DP-johto. Tällöin S on toteutumaton jos ja vain jos DP-johto $R_1, \ldots, R_n$ on onnistunut.

Todistus. Muunnossäännöt säilyttävät toteutuvuuden, joten kaikille $1 < i \leq n$ pätee: rintama R_i on (klausuulijoukkojensa disjunktiona) toteutuva jos ja vain jos R_{i-1} on toteutuva.

Näinollen R_1 on toteutumaton jos ja vain jos R_n on toteutumaton. Väite seuraa, koska $R_1 = [S]$ ja R_n on toteutumaton jos ja vain jos $R_1, \ldots, R_n$ on onnistunut.

Esimerkki. Oletetaan, että klausuulijoukosta S saadaan haarautumissäännöllä S_L ja $S_{\overline{L}}$. Nyt S on toteutuva jos ja vain jos S_L on toteutuva tai $S_{\overline{L}}$ on toteutuva.

DP-todistukset

- Lauseen P *DP-todistus* saadaan muuntamalla $\neg P$ klausuulijoukoksi S ja etsimällä rintamasta $R_1 = [S]$ lähtevä valmis ja onnistunut DP-johto $R_1, \ldots, R_n$.
- Epäonnistunut DP-johto antaa klausuulijoukolle mallin:

  Otetaan malli, joka täyttää tyhjään klausuulijoukkoon johtaneen muunnossääntöketjun yhteydessä saadut malliehdot.

Teoreema 3 (DP-todistusten virheettömyys ja täydellisyys)
Lause P on pätevä joss lauseella P on DP-todistus.

Todistus.

P pätevä	joss	lauseen $\neg P$ klausuulimuoto S on toteutumaton
	joss	rintamasta $R_1 = \{S\}$ voidaan muodostaa valmis ja onnistunut DP-johto $R_1, \ldots, R_n$
	joss	lauseella P on DP-todistus.

Esimerkki.

$$\begin{aligned}\neg((P \to Q) \to (\neg Q \to \neg P)) &\rightsquigarrow (P \to Q) \wedge \neg(Q \vee \neg P)\\ &\rightsquigarrow (\neg P \vee Q) \wedge \neg Q \wedge P\\ &\rightsquigarrow S = \{\neg P \vee Q, \neg Q, P\}\end{aligned}$$

Lauseen DP-todistus:

$$[\{\neg P \vee Q, \neg Q, P\}]$$
$$[\{P, \neg P\}]$$
$$[\{\bot\}]$$

DP-taulut

- Rintamien käyttö voidaan ymmärtää leveyshakuna.
- Jos haarautumissääntöä käytetään paljon, saattaa rintamien tilavaatimus muodostua eksponentiaaliseksi.
- Algoritmien tilankäyttö saadaan polynomiseksi, jos onnistuneen DP-johdon haku suoritetaan syvyyshakuna.
- Syntyvä hakupuu on muodoltaan binäärinen (ainoastaan haarautumissääntö haarauttaa puun).
- Ainoastaan yhtä hakupuun haaraa tarvitsee kerrallaan säilyttää muistissa (pino on tähän erinomaisesti soveltuva tietorakenne).
- Määrittelemme jatkossa binääristä hakupuuta vastavaan todistuksen käsitteen.

Määritelmä 4
Klausuulijoukoista muodostuva binääripuu T on DP-taulu, jos

(i) T:n ainoana solmuna (eli sekä juuri- että lehtisolmuna) on mikä tahansa klausuulijoukko S, tai

(ii) 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.

- Jokaista DP-taulua T vastaa rintama $R(T)$, joka on T:n lehtisolmuina olevien klausuulijoukkojen S joukko.
- DP-taulu T on valmis/onnistunut/epäonnistunut jos ja vain jos rintama $R(T)$ on valmis/onnnistunut/epäonnistunut.

Esimerkki. Todistetaan $((p \rightarrow q) \wedge (q \rightarrow p) \wedge (\neg p \rightarrow q)) \rightarrow (p \wedge q)$.

(i) Muunnetaan lauseen negaatio $\neg(((p \rightarrow q) \wedge (q \rightarrow p) \wedge (\neg p \rightarrow q)) \rightarrow (p \wedge q))$ klausuulimuotoon $\{\neg p \vee q, \neg q \vee p, p \vee q, \neg p \vee \neg q\}$.

(ii) Haetaan onnistunut DP-johto (esitetään klausuulijoukkojen joukkoa puuna; malliehdot suluissa).

$\{\neg p \vee q, \neg q \vee p,$
$p \vee q, \neg p \vee \neg q\}$

(p)	$(\neg p)$
$\{q, \neg q\}$	$\{\neg q, q\}$
(q)	$(\neg q)$
$\{\bot\}$	$\{\bot\}$

Esimerkki. Tutkitaan, onko lause $((r \rightarrow p \vee q) \wedge (p \vee q) \wedge (r \rightarrow \neg q) \wedge (q \vee s) \wedge (s \rightarrow r)) \rightarrow (s \wedge r)$ pätevä.

(i) Lauseen negaatio klausuuleina $\{\neg r \vee p \vee q, p \vee q, \neg r \vee \neg q, q \vee s, \neg s \vee r, \neg s \vee \neg r\}$

(ii) Haetaan DP-johtoa.

(iii) Epäonnistunut haara (klausuulijoukko tyhjä) antaa alkuperäiselle klausuulijoukolle mallin $\{p, q\}$.

(iv) Koska klausuulijoukko on ekvivalentti alkuperäisen lauseen negaation kanssa, alkuperäiselle lauseelle saadaan vastamalli.

$\{\neg r \vee p \vee q, p \vee q, \neg r \vee \neg q,$
$q \vee s, \neg s \vee r, \neg s \vee \neg r\}$
(peittosääntö)
$\{p \vee q, \neg r \vee \neg q, q \vee s,$
$\neg s \vee r, \neg s \vee \neg r\}$
(p)
$\{\neg r \vee \neg q, q \vee s, \neg s \vee r, \neg s \vee \neg r\}$

$(\neg r)$	(r)
$\{q \vee s, \neg s\}$	$\{\neg q, q \vee s, \neg s\}$
$(\neg s)$	$(\neg q)$
$\{q\}$	$\{s, \neg s\}$
(q)	(s)
$\{\}$	$\{\bot\}$

Davis-Putnam vs. taulumenetelmä

- Taulumenetelmässä ei ole cut-sääntöä: $\dfrac{}{P \mid \neg P}$
- Davis-Putnam -menetelmällä voidaan polynomisesti simuloida taulutodistus klausuulijoukolle.

$$\frac{L_1 \vee C_2}{L_1 \mid C_2} \mapsto \begin{array}{c|c} \multicolumn{2}{c}{\{\ldots, L_1 \vee C_2, \ldots\}} \\ (L_1) & (\overline{L_1}) \\ \{\ldots, \ldots\} & \{\ldots, C_2, \ldots\} \end{array}$$

- Taulumenetelmällä ei voi simuloida polynomisesti DP-johtoa. Esim.: $\{\neg p \vee q, \neg q \vee p, p \vee q, \neg p \vee \neg q\}$

☞ Davis-Putnam -menetelmä on vahvempi todistusmenetelmä kuin taulut ilman cut-sääntöä.

Toteutustekniikkaa

Yleensä syvyyshaku klausuulijoukottain, kunnes löytyy malli tai $\bot$.

$\mathrm{DP}(S, M)$
$\langle S', M' \rangle := \mathrm{simplify}(S, M)$;
if $S' = \emptyset$ **then** return M'
else if $\bot \in S'$ **then** return 'UNSAT'
else
  $L := \mathrm{choose}(S', M')$;
  $M'' := \mathrm{DP}(S' \cup \{L\}, M' \cup \{L\})$;
  if M'' = 'UNSAT' **then** return $\mathrm{DP}(S' \cup \{\overline{L}\}, M' \cup \{\overline{L}\})$
  else return M''
  endif
endif

- $\mathrm{simplify}(S, M)$

  Yksinkertaistaa klausuulijoukkoa käyttäen muunnossääntöjä. Palauttaa yksinkertaistetun klausuulijoukon ja malliehdot.

  Yleensä ainakin yhden literaalin sääntö.
  Joskus komplementti puuttuu -sääntö.
  Peittosääntö ei yleensä käytössä.

- $\mathrm{choose}(S, M)$

  Heuristiikka, jolla valitaan atomilause A haarautumissääntöön ja se kumpi haara $(A/\neg A)$ käydään läpi ensin.

  Tyypillinen idea: suositaan atomilauseita, jotka esiintyvät usein lyhyimmissä klausuuleissa.

  ☞ Heuristiikalla ratkaiseva vaikutus suorituskykyyn.

Esimerkki: Crawfordin ntab

$\mathrm{simplify}(S, M)$:

- Laskee joukosta S yksikköresoluutiolla (unit resolution):

  $$\frac{L, \overline{L} \vee L_1 \vee \cdots \vee L_n}{L_1 \vee \cdots \vee L_n}$$

  saatavat klausuulit.

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

- Yksikköresoluutiolla saatavat klausuulit voidaan laskea lineaarisessa ajassa.

- Lopetusehto: kaikilla atomilauseilla totuusarvo (malliehto).

$\mathrm{choose}(S, M)$:

- (i) Lasketaan kullekin atomilauseelle A hyvyysluku

  $$\mathrm{score}(A) = \mathrm{pc}(A) * \mathrm{nc}(A) * 1024 + \mathrm{pc}(A) + \mathrm{nc}(A) + 1 \qquad (1)$$

  missä $\mathrm{pc}(A)$ $(\mathrm{nc}(A))$ on binääriklausuulien määrä, joissa A esiintyy positiivisena (negatiivisena).

- (ii) Valitaan k parasta atomilausetta ($k = v - 21 * va$ missä v on atomilauseiden määrä ja va on niiden atomilauseiden määrä, jolla on jo totuusarvo)

- (iii) Lasketaan kullekin atomilauseelle A uusi hyvyysluku kaavalla (1) missä nyt $\mathrm{pc}(A)$ $(\mathrm{nc}(A))$ kertoo, montako uutta binääriklausuulia saadaan yksikköresoluutiolla, kun joukkoon S lisätään A $(\neg A)$. Valitaan atomilause, jolla on paras ko. hyvyysluku.

Muita sääntöjä

- Literaali epäonnistuu -sääntö:

  Jos tyhjä klausuuli saadaan joukosta $S \cup \{L\}$ yksikköresoluutiolla, poistetaan kaikki klausuulit, joissa esiintyy $\overline{L}$ ja muista klausuuleista literaali L.

  (Malliehto: $\overline{L}$).

  Esimerkki. Olkoon $S = \{A \vee B, \neg B \vee C, \neg C \vee A\}$.
  Joukosta $S \cup \{\neg A\}$ saadaan yksikköresoluutiolla $\{B, C, \neg C, \perp\}$.
  Siis saadaan muunnettu klausuulijoukko $S = \{\neg B \vee C\}$ (malliehto: A).

- Literaalipari epäonnistuu -sääntö:
  Jos tyhjä klausuuli saadaan joukosta $S \cup \{L_1, L_2\}$ yksikköresoluutiolla, lisätään joukkoon S klausuuli $\overline{L_1} \vee \overline{L_2}$.
  Esimerkki. Olkoon $S = \{A \vee \neg B \vee C, \neg C \vee A\}$.
  Joukosta $S \cup \{\neg A, B\}$ saadaan yksikköresoluutiolla $\{C, A, \bot\}$.
  Joten joukkoon S lisätään klausuuli $A \vee \neg B$.
- 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 $C_0 \vee C_1, \ldots, C_0 \vee C_k$.

Arviointia

- Davis-Putnam -menetelmä on yksinkertainen toteuttaa tehokkaasti.
- Hakuavaruuden koko 2^v, missä v on klausuulijoukon atomilauseiden määrä.
- Tyypillisesti polynominen tilavaativuus.
- Kussakin hakuavaruuden pisteessä
  (a) yksinkertaistetaan klausuulijoukkoa
  (b) valitaan uusi atomilause haarautumista varten
  (hakuheuristiikka)
- Trade-off: mitä enemmän yksinkertaistetaan sitä vähemmän haarautumista.

- Tyypillinen ratkaisu:
  1. Tehdään esikäsittely, jossa yksinkertaistetaan klausuulijoukkoa ja vähennetään muuttujien määrää.
     Esim. käyttämällä yhden literaalin, komplementti puuttuu, literaali epäonnistuu ja poista 1-literaalit sääntöä.
     (polynominen aikavaativuus)
  2. Kussakin hakupisteessä yksinkertaistaminen lineaarisessa ajassa (yksikköresoluutio).
  3. Kussakin hakupisteessä heuristiikka lineaarisessa ajassa (atomilauseiden esiintymistiheys/yksikköresoluutio).
  4. Ei käytetä monimutkaisia peruutusmenetelmiä.