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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Perusidea

- Tutkitaan klausuulijoukkojen joukon toteutuvuutta:

  Klausuulijoukkojen joukko on toteutuva $\Longleftrightarrow$ 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, se on toteutuva myös sen jälkeen.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

- **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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

- **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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

- **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 $S = \{A \lor \neg B \lor \neg C,\ \neg A \lor \neg B\}$ saadaan:

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

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

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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 joukoon S' yhtä edellä kuvatuista säännöistä.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

DP-johtojen ominaisuuksia

Rintama R on

- *valmis*, jos sen jokaiselle klausuulijoukolle $S \in R$ pätee *joko* $\perp \in S$ *tai* $S = \emptyset$.
- *onnistunut*, jos sen jokaiselle klausuulijoukolle $S \in R$ pätee $\perp \in S$.
- *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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Onnistunut DP-johto:

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

Esimerkki. Epäonnistunut DP-johto:

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

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Väite. (DP-johtojen 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.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Väite. (DP-johtojen ominaisuudet II)
Olkoon S äärellinen klausuulijoukko ja $R_1, \ldots, R_n$ rintamasta $R_1 = [S]$ modostettu valmis DP-johto. Tällöin S on toteutumaton $\Longleftrightarrow$ 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 $\Longleftrightarrow$ R_{i-1} on toteutuva.

Näinollen R_1 on toteutumaton $\Longleftrightarrow$ R_n on toteutumaton.

Väite seuraa, koska $R_1 = [S]$ ja R_n on toteutumaton $\Longleftrightarrow$ DP-johto $R_1, \ldots, R_n$ on onnistunut.

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

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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. DP-todistusten *virheettömyys* ja *täydellisyys:*
Lause P on pätevä $\iff$ lauseella P on DP-todistus.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Todistus.

P pätevä $\iff$ lauseen $\neg P$ klausuulimuoto S on toteutumaton

$\iff$ rintamasta $R_1 = \{S\}$ voidaan muodostaa valmis ja onnistunut DP-johto $R_1, \ldots, R_n$

$\iff$ lauseella P on DP-todistus.

Esimerkki.

$$\begin{aligned} \neg((P \to Q) \to (\neg Q \to \neg P)) &\leadsto (P \to Q) \land \neg(Q \lor \neg P) \\ &\leadsto (\neg P \lor Q) \land \neg Q \land P \\ &\leadsto S = \{\neg P \lor Q, \neg Q, P\} \end{aligned}$$

Lauseen DP-todistus:

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

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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).

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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.

- 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/onnistunut/epäonnistunut.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Todistetaan lause $\phi =$

$$((P \rightarrow Q) \wedge (Q \rightarrow P) \wedge (\neg P \rightarrow Q)) \rightarrow (P \wedge Q).$$

1. Muunnos klausuulimuotoon:

$$\begin{aligned} \neg\phi &\rightsquigarrow (P \rightarrow Q) \wedge (Q \rightarrow P) \wedge (\neg P \rightarrow Q) \wedge \neg(P \wedge Q) \\ &\rightsquigarrow \{\neg P \vee Q,\ \neg Q \vee P,\ P \vee Q,\ \neg P \vee \neg Q\}. \end{aligned}$$

2. Muodostetaan onnistunut DP-taulu (malliehdot annettu suluissa):

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

$$\begin{array}{c|c} (P) & (\neg P) \\ \{Q, \neg Q\} & \{\neg Q, Q\} \\ (Q) & (\neg Q) \\ \{\bot\} & \{\bot\} \end{array}$$

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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 $\mathcal{A}$.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

➤ DP-taulusta muodostuu seuraava:

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

(peittosääntö)

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

$$(P)$$

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

$$\begin{array}{c|c}
(\neg R) & (R) \\
\{Q \vee S,\ \neg S\} & \{\neg Q,\ Q \vee S, \neg S\} \\
(\neg S) & (\neg Q) \\
\{Q\} & \{S, \neg S\} \\
(Q) & (S) \\
\{\} & \{\perp\}
\end{array}$$

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

DP-taulut vs. semanttiset taulut

➤ Semanttisissa tauluissa ei käytetä haarautumissääntöä.

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

$$\begin{array}{ccccc}
\multicolumn{2}{c}{T(L_1 \vee C_2)} & & \multicolumn{2}{c}{\{\dots, L_1 \vee C_2, \dots\}} \\
/ & \backslash & \mapsto & (L_1) & \multicolumn{1}{|c}{(\overline{L_1})} \\
TL_1 & TC_2 & & \{\dots, \dots\} & \multicolumn{1}{|c}{\{\dots, C_2, \dots\}}
\end{array}$$

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

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

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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 $\{B, C, \neg C, \bot\}$.

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

Malliehto: A.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

- **Literaalipari epäonnistuu -sääntö:**

  Jos klausuulijoukosta $S \cup \{L_1, L_2\}$ saadaan yksikköresoluutiolla $\bot$, 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 $S = \{A \vee \neg B \vee C, \neg C \vee A\}$.
Joukosta $S \cup \{\neg A, B\}$ saadaan yksikköresoluutiolla $\{C, A, \bot\}$.

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

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio