# Ratkaisumenetelmät

- Temporaalilogiikat (CTL/LTL) ratkevia, koska niillä on äärellisen mallin ominaisuus (vastamallin koolle yläraja), mutta tehokkaampia ratkaisumenetelmiä saadaan taulujen ja automaattiteorian avulla.
- Taulumenetelmä (esim. CTL)
  (i) Rakennetaan tutkittavalle lauseelle taulugraafi, joka sisältää (oleellisesti) kaikki lauseen mahdolliset mallit.
  (ii) Karsitaan taulua ja tarkastetaan, jäikö siihen tutkivan lauseen malleja.
- Automaattiteoreettinen menetelmä (esim. LTL)
  (i) Rakennetaan tutkittavalle lauseelle äärellistilainen (Büchi-) automaatti, joka hyväksyy äärettömiä objekteja (esim. jonoja) siten, että automaatti hyväksyy (oleellisesti) kaikki lauseen mallit.
  (ii) Tarkistetaan, onko automaatin hyväksymä kieli tyhjä.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



# Taulumenetelmä CTL:lle

CTL:n taulu on bipartiitti graafi, jossa on solmut ovat lausejoukkoja ja niitä on kahdentyyppisiä: OR-solmuja ja AND-solmuja.

Lauseelle $P$ taulu rakennetaan muuntamalla $P$ ensin *positiiviseen normaalimuotoon* (jossa negaatio esiintyy vain atomilauseen edessä) ja etenemällä sitten kahdessa vaiheessa:

(i) rakennetaan alustava taulu $T_0$ ja

(ii) muodostetaan lopullinen taulu $T_1$ karsintasäännöillä taulusta $T_0$.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



CTL-lause muunnetaan **positiiviseen normaalimuotoon** seuraavilla säännöillä:

$$
\begin{array}{lcl}
P \to Q & \mapsto & \neg P \vee Q \\
\neg(P \vee Q) & \mapsto & \neg P \wedge \neg Q \\
\neg(P \wedge Q) & \mapsto & \neg P \vee \neg Q \\
\neg\neg P & \mapsto & P \\
\neg\mathbf{AG}P & \mapsto & \mathbf{EF}\neg P \\
\neg\mathbf{EF}P & \mapsto & \mathbf{AG}\neg P \\
\neg\mathbf{EG}P & \mapsto & \mathbf{AF}\neg P \\
\neg\mathbf{AF}P & \mapsto & \mathbf{EG}\neg P \\
\neg\mathbf{AX}P & \mapsto & \mathbf{EX}\neg P \\
\neg\mathbf{EX}P & \mapsto & \mathbf{AX}\neg P
\end{array}
\qquad
\begin{array}{lcl}
\neg\mathbf{A}(P\mathbf{U}Q) & \mapsto & \mathbf{E}(\neg P\mathbf{B}Q) \\
\neg\mathbf{E}(P\mathbf{U}Q) & \mapsto & \mathbf{A}(\neg P\mathbf{B}Q) \\
\neg\mathbf{A}(P\mathbf{B}Q) & \mapsto & \mathbf{E}(\neg P\mathbf{U}Q) \\
\neg\mathbf{E}(P\mathbf{B}Q) & \mapsto & \mathbf{A}(\neg P\mathbf{U}Q)
\end{array}
$$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



- Huom! Lyhennysmerkinnät:
  $\mathbf{A}(P\mathbf{B}Q)$: $\neg\mathbf{E}((\neg P)\mathbf{U}Q)$
  $\mathbf{E}(P\mathbf{B}Q)$: $\neg\mathbf{A}((\neg P)\mathbf{U}Q)$
- Positiivisessa normaalimuodossa käytetään konnektiiveja $\wedge, \vee$ ja negaatioita esiintyy vain atomilauseiden edessä.
- Merkintä $\sim P$: lauseen $\neg P$ positiivinen normaalimuoto.

  **Esimerkki.**

  $\sim\mathbf{AG}(R \to (\neg Q \wedge \mathbf{A}(P\mathbf{U}Q))) : \mathbf{EF}(R \wedge (Q \vee \mathbf{E}(\neg P\mathbf{B}Q)))$

  koska $\neg\mathbf{AG}(R \to (\neg Q \wedge \mathbf{A}(P\mathbf{U}Q)))$

  $\mapsto \mathbf{EF}\neg(\neg R \vee (\neg Q \wedge \mathbf{A}(P\mathbf{U}Q)))$

  $\mapsto \mathbf{EF}(R \wedge \neg(\neg Q \wedge \mathbf{A}(P\mathbf{U}Q)))$

  $\mapsto \mathbf{EF}(R \wedge (Q \vee \neg\mathbf{A}(P\mathbf{U}Q)))$

  $\mapsto \mathbf{EF}(R \wedge (Q \vee \mathbf{E}(\neg P\mathbf{B}Q)))$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

## Alustava taulu

Alustavan taulun $T_0$ muodostaminen:

- Aloitetaan OR-solmusta $D_0 = \{P\}$.
- OR-solmun $D$ seuraajat ovat AND-solmuja, jotka saadaan soveltamalla *$\alpha/\beta$-sääntöjä* solmuun $D$.
- AND-solmun $C$ seuraajat ovat OR-solmuja, jotka saadaan *seuraajasäännöllä* solmusta $C$.
- Huom! Jos OR-solmun $D$ seuraaja $C$ esiintyy jo taulussa, ei tauluun tule toista kopiota, vaan $D$:n seuraajaksi tulee jo olemassa oleva solmu $C$. Samoin AND-solmuille.

  $\boxed{\Rightarrow}$ Taulu äärellinen

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



## OR-solmun seuraajat

OR-solmun $D$ seuraajat ovat $D$:sta seuraavilla *$\alpha$- ja $\beta$-säännöillä* saatavat alaspäin suljetut joukot.
Joukko $C$ on alaspäin suljettu, kun seuraavat kaksi ehtoa toteutuvat:
(i) Jos $\alpha \in C$, $\alpha_1 \in C$ ja $\alpha_2 \in C$.
(ii) Jos $\beta \in C$, $\beta_1 \in C$ tai $\beta_2 \in C$.

**$\alpha$-säännöt:**

$$\frac{P \wedge Q}{\begin{array}{c} P \\ Q \end{array}} \qquad \frac{\mathbf{AG}P}{\begin{array}{c} P \\ \mathbf{AXAG}P \end{array}} \qquad \frac{\mathbf{EG}P}{\begin{array}{c} P \\ \mathbf{EXEG}P \end{array}}$$

$$\frac{\mathbf{A}(P\mathbf{B}Q)}{\begin{array}{c} \sim Q \\ P \vee \mathbf{AXA}(P\mathbf{B}Q) \end{array}} \qquad \frac{\mathbf{E}(P\mathbf{B}Q)}{\begin{array}{c} \sim Q \\ P \vee \mathbf{EXE}(P\mathbf{B}Q) \end{array}}$$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



**$\beta$-säännöt:**

$$\frac{P \vee Q}{P \mid Q} \qquad \frac{\mathbf{AF}P}{P \mid \mathbf{AXAF}P} \qquad \frac{\mathbf{EF}P}{P \mid \mathbf{EXEF}P}$$

$$\frac{\mathbf{A}(P\mathbf{U}Q)}{Q \mid P \wedge \mathbf{AXA}(P\mathbf{U}Q)} \qquad \frac{\mathbf{E}(P\mathbf{U}Q)}{Q \mid P \wedge \mathbf{EXE}(P\mathbf{U}Q)}$$

Huom! Literaaleille sekä $\mathbf{AX}P$ ja $\mathbf{EX}P$ lauseille ei sääntöjä.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



**Esimerkki.**

Olkoon $D = \{\mathbf{AFEF}((P \wedge Q) \vee R)\}$
$D$:sta saatavat alaspäin suljetut joukot ovat:

$C_1 = \{\mathbf{AFEF}((P \wedge Q) \vee R), \mathbf{EF}((P \wedge Q) \vee R), (P \wedge Q) \vee R, P \wedge Q, P, Q\}$

$C_2 = \{\mathbf{AFEF}((P \wedge Q) \vee R), \mathbf{EF}((P \wedge Q) \vee R), (P \wedge Q) \vee R, R\}$

$C_3 = \{\mathbf{AFEF}((P \wedge Q) \vee R), \mathbf{EF}((P \wedge Q) \vee R), \mathbf{EXEF}((P \wedge Q) \vee R)\}$

$C_4 = \{\mathbf{AFEF}((P \wedge Q) \vee R), \mathbf{AXAFEF}((P \wedge Q) \vee R)\}$

(Alaspäin suljetut joukot voidaan rakentaa lauselogiikan taulumenetelmän tapaan.)

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

## AND-solmun seuraajat

AND-solmun $C$ seuraajat saadaan *seuraajasäännöllä*:

Olkoon joukossa $C$ seuraavatilalauseet:
$\mathbf{AX}P_1, \ldots, \mathbf{AX}P_l$,
$\mathbf{EX}Q_1, \ldots, \mathbf{EX}Q_k$.
Tällöin $C$:llä on seuraajat:
$D_1 = \{P_1, \ldots, P_l, Q_1\} \ldots$
$D_k = \{P_1, \ldots, P_l, Q_k\}$

**Esimerkki.** Olkoon
$C = \{\mathbf{A}(P\mathbf{U}Q), \mathbf{AXA}(P\mathbf{U}Q), \mathbf{EG}P, P, \mathbf{EXEG}P, \mathbf{EF}Q, \mathbf{EXEF}Q\}$.
Tällöin $C$:n seuraajat ovat
$D_1 = \{\mathbf{A}(P\mathbf{U}Q), \mathbf{EG}P\}$ ja $D_2 = \{\mathbf{A}(P\mathbf{U}Q), \mathbf{EF}Q\}$.
(Huom! Kullakin AND-solmulla pitää olla ainakin yksi seuraaja.)

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



## Karsintasäännöt

Alustavasta taulusta $T_0$ saadaan lopullinen taulu suorittamalla karsintaa seuraavilla säännöillä, kunnes mitään niistä ei voida soveltaa.

- Poistetaan AND-solmu, joka sisältää lauseen ja sen negaation.
- Poistetaan AND-solmu, jos yksikin sen alkuperäisistä seuraajista on poistettu.
- Poistetaan OR-solmu, jos kaikki sen alkuperäisistä seuraajista on poistettu.
- Poistetaan AND-solmu, jos jokin sen **tulevaisuuslause** ei toteudu tämän hetkisessä taulussa (mallintarkastus).

Tulevaisuuslauseita ovat seuraavaa muotoa olevat lauseet:

$\mathbf{E}(P\mathbf{U}Q)$, $\mathbf{EF}Q$, $\mathbf{A}(P\mathbf{U}Q)$, $\mathbf{AF}Q$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



Tulevaisuuslauseiden toteutuminen taulussa määritellään seuraavasti:

- Tulevaisuuslause $\mathbf{EF}Q$ ($\mathbf{E}(P\mathbf{U}Q)$) toteutuu AND-solmussa $C$, joss taulusta löytyy solmusta $C$ lähtevä polku AND-solmuun $C'$, joka sisältää lauseen $Q$ (ja polun kaikissa muissa AND-solmuissa on lause $P$).
- Tulevaisuuslause $\mathbf{AF}Q$ ($\mathbf{A}(P\mathbf{U}Q)$) toteutuu AND-solmussa $C$, joss taulusta löytyy asyklinen aligraafi, jolle pätee:
  (i) Aligraafin juuri on solmu $C$.
  (ii) Kullekin OR-sisäsolmulle täsmälleen yksi sen AND-seuraajasolmu on aligraafissa.
  (iii) Kullekin AND-sisäsolmulle kaikki sen OR-seuraajasolmut ovat mukana aligraafissa.
  (iv) Kaikissa aligraafin lehtisolmuissa on lause $Q$ (ja kaikissa muissa AND-solmuissa $P$).

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

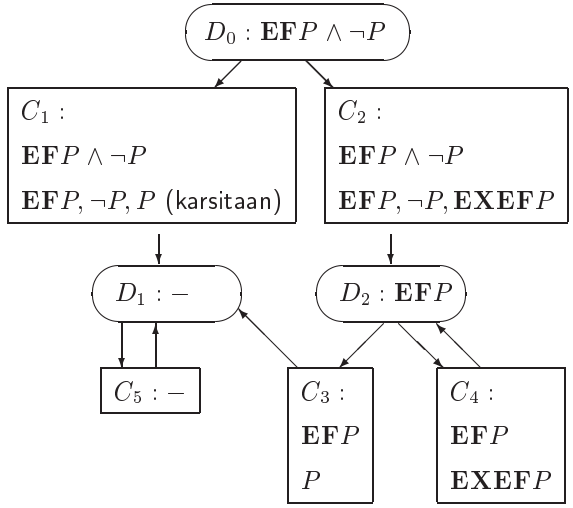


**Esimerkki.** Olkoon $D_0 = \{\mathbf{EF}P \wedge \neg P\}$.

- $D_0$:n AND-seuraajat:
  $C_1 = \{\mathbf{EF}P \wedge \neg P, \mathbf{EF}P, \neg P, P\}$ ja
  $C_2 = \{\mathbf{EF}P \wedge \neg P, \mathbf{EF}P, \neg P, \mathbf{EXEF}P\}$
- $C_1$:n OR-seuraajat: $D_1 = \{\}$ (Huom. väh. yksi seuraaja.)
  $C_2$:n OR-seuraajat: $D_2 = \{\mathbf{EF}P\}$
- $D_2$:n AND-seuraajat: $C_3 = \{\mathbf{EF}P, P\}$ ja $C_4 = \{\mathbf{EF}P, \mathbf{EXEF}P\}$
- $D_1$:n AND-seuraajat: $C_5 = \{\}$
- $C_3$:n OR-seuraajat: $\{\} = D_1$
  $C_4$:n OR-seuraajat: $\{\mathbf{EF}P\} = D_2$
  $C_5$:n OR-seuraajat: $\{\} = D_1$
- Alustava taulu $T_0$ on valmis.
- Karsinta: Poistetaan $C_1$ (lause ja negaatio). Lopullinen taulu $T_1$ valmis. (Ainoa tulev. lause $\mathbf{EF}P$ toteutuu jok. AND-solmussa.)

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Yleensä taulut tehdään graafin muotoon:




© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



## CTL-toteutuvuus tauluilla

**Teoreema.** Olkoon CTL-lause $P$ positiivisessa normaalimuodossa. Tällöin $P$ on toteutuva joss lopullisessa taulussa $T_1$ on AND-solmu, joka sisältää lauseen $P$.

- CTL-taulu antaa lauseen $P$ mallin:
  Mallin tilat AND-solmuja ja valuaatio suoraan AND-solmun sisältämistä atomilauseista.
  Seuraajat valittava siten, että tulevaisuuslauseet toteutuvat.
- Taulumenetelmä ohjelmasynteesissä:
  (i) Ohjelman määrittely CTL-lauseina.
  (ii) Tauluilla määrittelylle automaattisesti malli, joka antaa ohjelman ohjausrakenteen.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



## LTL-toteutuvuus tauluilla

CTL-taulut antavat menetelmän myös LTL-toteutuvuuden tarkastamiseen:

**Teoreema.** Olkoon LTL-lause $P$ positiivisessa normaalimuodossa ja CTL-lause $P'$ saatu siitä korvaamalla operaattorit $\mathbf{F}, \mathbf{G}, \mathbf{X}, \mathbf{U}, \mathbf{B}$, operaattoreilla $\mathbf{AF}, \mathbf{AG}, \mathbf{AX}, \mathbf{AU}, \mathbf{AB}$. Tällöin $P$ on LTL-toteutuva joss $P'$ on CTL-toteutuva.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



**Esimerkki.** Anna CTL-lauseelle $\mathbf{EF}P \wedge \neg P$ malli.

Muodostetaan malli $\langle S, R, v \rangle$ edellä annetusta taulusta $T_1$ esimerkiksi seuraavasti:

$S = \{C_2, C_3, C_5\}$

$R = \{\langle C_2, C_3 \rangle, \langle C_3, C_5 \rangle, \langle C_5, C_5 \rangle\}$

$v(P, s) = \text{true}$ jos $s = C_3$ muutoin $v(P, s) = \text{false}$

Toinen vaihtoehto:

$S = \{C_2, C_3, C_4, C_5\}$

$R = \{\langle C_2, C_4 \rangle, \langle C_4, C_3 \rangle, \langle C_3, C_5 \rangle, \langle C_5, C_5 \rangle\}$

$v(P, s) = \text{true}$ jos $s = C_3$ muutoin $v(P, s) = \text{false}$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

**Esimerkki.** Onko CTL-lause $\varphi$:

$\mathbf{EX}(P \vee Q) \rightarrow (\mathbf{EX}P \vee \mathbf{EX}Q)$

pätevä?

- Lause on pätevä joss sen negaatio ei ole toteutuva.
- Toteutuvuus voidaan selvittää taulumenetelmällä:
  (i) Muodostetaan lauseen $\varphi$ **negaation** positiivinen normaalimuoto $\sim\varphi$.
  (ii) Rakennetaan taulu lauseelle $\sim\varphi$.
- $\varphi$ on pätevä joss $\sim\varphi$ ei ole toteutuva joss lopullisessa taulussa ei ole yhtään AND-solmua, joka sisältää lauseen $\sim\varphi$.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



**Esimerkki–jatkuu.**

Lauseen $\neg\varphi$ positiivinen normaalimuoto $\sim\varphi$:

$\neg(\mathbf{EX}(P \vee Q) \rightarrow (\mathbf{EX}P \vee \mathbf{EX}Q)$

$\mapsto \quad \mathbf{EX}(P \vee Q) \wedge \neg(\mathbf{EX}P \vee \mathbf{EX}Q)$

$\mapsto \quad \mathbf{EX}(P \vee Q) \wedge (\neg\mathbf{EX}P \wedge \neg\mathbf{EX}Q)$

$\mapsto \quad \mathbf{EX}(P \vee Q) \wedge (\mathbf{AX}\neg P \wedge \mathbf{AX}\neg Q)$

$\sim\varphi$: $\mathbf{EX}(P \vee Q) \wedge (\mathbf{AX}\neg P \wedge \mathbf{AX}\neg Q)$

Taulu: $D_0 = \{\sim\varphi\}$.

- $D_0$:n AND-seuraajat:
  $C_1 = \{\sim\varphi, \mathbf{EX}(P \vee Q), \mathbf{AX}\neg P, \mathbf{AX}\neg Q\}$
- $C_1$:n OR-seuraajat: $D_1 = \{P \vee Q, \neg P, \neg Q\}$
- $D_1$:n AND-seuraajat:
  $C_2 = D_1 \cup \{P\}$ ja $C_3 = D_1 \cup \{Q\}$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



- $C_2$:n OR-seuraajat: $D_3 = \{\}$
  $C_3$:n OR-seuraajat: $\{\} = D_3$
  $D_3$:n AND-seuraajat: $C_4 = \{\}$
  $C_4$:n OR-seuraajat: $\{\} = D_3$
- Alustava taulu $T_0$ on valmis.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio



- Karsinta:
  (i) $C_2, C_3$ voidaan poistaa (sisältävät lauseen ja sen negaation)
  (ii) $D_1$ voidaan poistaa (ei seuraajia)
  (iii) $C_1$ voidaan poistaa (seuraaja poistettu)
  (iv) $D_0$ voidaan poistaa (ei seuraajia)

  Lopullinen taulu $T_1$ valmis.

  Taulussa $T_1$ ei ole AND-solmua, jossa esiintyy $\sim\varphi$.

  $\boxed{\Rightarrow}$ $\sim\varphi$ ei ole toteutuva. Siis $\varphi$ on pätevä.

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

# Laskennallinen vaativuus

- CTL
  Mallintarkastus: $\mathbf{P}$-täydellinen
  $O(|M| \cdot |P|)$
  Toteutuvuus: **EXPTIME**-täydellinen
- LTL
  Mallintarkastus: **PSPACE**-täydellinen
  $O(|M| \cdot exp(|P|))$
  Toteutuvuus: **PSPACE**-täydellinen
- CTL$^*$
  Mallintarkastus: **PSPACE**-täydellinen
  $O(|M| \cdot exp(|P|))$
  Toteutuvuus: 2**EXPTIME**-täydellinen

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

# Tehokkaasti toteutettavia aliluokkia

- Toteutuvuusongelma ratkeaa polynomisessa ajassa esim. ohjelmasynteesin kannalta mielenkiintoisissa tapauksissa.
- Esim. SCTL (yksinkertaistettu CTL)
  $P_1 \vee \cdots \vee P_n$
  $\mathbf{AG}(P_1 \vee \cdots \vee P_n)$
  $\mathbf{AG}(P_0 \rightarrow \mathbf{AF}(P_1 \vee \cdots \vee P_n))$
  $\mathbf{AG}(P_0 \rightarrow \mathbf{A}(P_1 \vee \cdots \vee P_n \mathbf{U} Q_1 \vee \cdots \vee Q_m))$
  $\mathbf{AG}(P_0 \rightarrow \mathbf{AX}(P_1 \vee \cdots \vee P_n) \wedge$
  $\mathbf{EX}(Q_1 \vee \cdots \vee Q_m) \wedge \cdots \wedge \mathbf{EX}(R_1 \vee \cdots \vee R_l))$
  jossa ESC-oletus pätee: tulevaisuuslauseet eivät riipu historiasta.
- Esim. RLTL (rajoitettu LTL): $\mathbf{G}(P_1 \vee \cdots \vee P_n)$
  $\mathbf{G}(P_0 \rightarrow \mathbf{F}(P_1 \vee \cdots \vee P_n))$
  $\mathbf{G}(P_0 \rightarrow \mathbf{X}(P_1 \vee \cdots \vee P_n))$

© 2002 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio