# RATKAISUMENETELMÄT

1. Taulumenetelmä CTL:lle
2. Toteutuvuuden ja pätevyyden tutkiminen
3. Laskennallinen vaativuus (yhteenveto)

E. A. Emerson: *Automated Temporal Reasoning about Reactive Systems*, luku 4 (s. 18–23).

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Taustaa

- Temporaalilogiikat (CTL/LTL) ovat ratkevia, koska niillä on äärellisen mallin ominaisuus
  (eli vastamallin koolle voidaan asettaa yläraja).
- Tehokkaampia ratkaisumenetelmiä saadaan semanttisten taulujen ja automaattiteorian avulla.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Keskeiset ratkaisustrategiat

- Taulumenetelmä (esim. CTL):
  (i) Rakennetaan tutkittavalle lauseelle taulugraafi, joka sisältää (oleellisesti) kaikki lauseen mahdolliset mallit.
  (ii) Karsitaan taulua ja tarkastetaan, jäikö siihen ko. lauseen malleja.
- Automaattiteoreettinen menetelmä (esim. LTL):
  (i) Rakennetaan tutkittavalle lauseelle äärellistilainen (Büchi-) automaatti, joka hyväksyy äärettömän pitkiä sanoja (polkuja) siten, että automaatti hyväksyy (oleellisesti) kaikki lauseen toteuttavat täydet polut.
  (ii) Tarkistetaan, onko automaatin hyväksymä kieli tyhjä.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## 1. Taulumenetelmä CTL:lle

- CTL:n taulu on bipartiitti graafi, jossa solmut ovat lausejoukkoja ja niitä on kahdentyyppisiä: OR-solmuja ja AND-solmuja.
- Lauseelle $P$ taulu rakennetaan muuntamalla $P$ ensin *positiiviseen normaalimuotoon* (jossa negaatiot esiintyvät vain atomilauseiden edessä) ja etenemällä sitten kahdessa vaiheessa:
  (i) rakennetaan alustava taulu $T_0$ ja
  (ii) muodostetaan lopullinen taulu $T_1$ karsintasäännöillä $T_0$:sta.
- Positiivisessa normaalimuodossa käytetään konnektiiveja $\wedge$, $\vee$ ja negaatioita $\neg$ esiintyy vain atomilauseiden edessä.
- Merkintä $\sim P$: lauseen $\neg P$ positiivinen normaalimuoto.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## Muunnossäännöt

CTL-lauseen positiivinen normaalimuoto muodostetaan seuraavasti:

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

Huom! Lyhennysmerkinnät:

$\mathbf{A}(P\mathbf{B}Q)$ : $\neg\mathbf{E}((\neg P)\mathbf{U}Q)$

$\mathbf{E}(P\mathbf{B}Q)$ : $\neg\mathbf{A}((\neg P)\mathbf{U}Q)$

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Esimerkki

Haetaan muunnossäännöillä positiivinen normaalimuoto $\sim\mathbf{AG}(R \rightarrow (\neg Q \wedge \mathbf{A}(P\mathbf{U}Q)))$:

$$
\begin{array}{rl}
 & \neg\mathbf{AG}(R \rightarrow (\neg Q \wedge \mathbf{A}(P\mathbf{U}Q))) \\
\mapsto & \mathbf{EF}\neg(\neg R \vee (\neg Q \wedge \mathbf{A}(P\mathbf{U}Q))) \\
\mapsto & \mathbf{EF}(R \wedge \neg(\neg Q \wedge \mathbf{A}(P\mathbf{U}Q))) \\
\mapsto & \mathbf{EF}(R \wedge (Q \vee \neg\mathbf{A}(P\mathbf{U}Q))) \\
\mapsto & \mathbf{EF}(R \wedge (Q \vee \mathbf{E}(\neg P\mathbf{B}Q)))
\end{array}
$$

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Alustavan taulun $T_0$ muodostaminen

- Aloitetaan OR-solmusta $D_0 = \{P\}$.
- OR-solmun $D$ seuraajat ovat AND-solmuja $C$, jotka saadaan soveltamalla $\alpha/\beta$-*sääntöjä* solmuun $D$.
- AND-solmun $C$ seuraajat ovat OR-solmuja $D$, jotka saadaan *seuraajasäännöllä* solmusta $C$.

**Huom.** Jos OR-solmun $D$ seuraaja (lausejoukko) $C$ esiintyy jo taulussa, ei tauluun tule toista kopiota, vaan $D$:n seuraajaksi tulee jo olemassa oleva solmu $C$ (vastaavasti AND-solmuille).

$\Longrightarrow$ Taulu muodostuu äärelliseksi.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## OR-solmun seuraajat

Kukin OR-solmun $D$ seuraaja $C$ on pienin joukko lauseita siten, että $D \subseteq C$ ja $C$ on suljettu alaspäin seuraavien $\alpha$- ja $\beta$-sääntöjen suhteen:

1. Jos $\alpha \in C$, niin $\alpha_1 \in C$ ja $\alpha_2 \in C$.
2. Jos $\beta \in C$, niin $\beta_1 \in C$ tai $\beta_2 \in C$.

**$\alpha$-säännöt:**

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

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## OR-solmun seuraajat (jatkoa)

$$\frac{\mathbf{A}(P\mathbf{B}Q)}{\sim Q \quad P \vee \mathbf{AXA}(P\mathbf{B}Q)} \qquad \frac{\mathbf{E}(P\mathbf{B}Q)}{\sim Q \quad P \vee \mathbf{EXE}(P\mathbf{B}Q)}$$

**β-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 muotoa $\mathbf{AX}P$ ja $\mathbf{EX}P$ oleville lauseille ei ole sääntöjä (seuraajan $C$ laajentaminen päättyy tällaisiin lauseisiin).

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Esimerkki

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

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

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

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

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

**Huom.** Alaspäin suljetut joukot voidaan rakentaa lauselogiikan taulumenetelmän tapaan. Kukin taulun haara vastaa yhtä joukkoa.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## AND-solmun seuraajat

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

- Jos joukossa $C$ on seuraajatilalauseet

  $$\mathbf{AX}P_1, \ldots, \mathbf{AX}P_l \text{ ja } \mathbf{EX}Q_1, \ldots, \mathbf{EX}Q_k,$$

  niin solmulla $C$ on seuraajat

  $$D_1 = \{P_1, \ldots, P_l, Q_1\}, \ldots, D_k = \{P_1, \ldots, P_l, Q_k\}.$$

- Jos joukossa $C$ ei ole muotoa $\mathbf{EX}Q_i$ olevia lauseita, joukolla on yksi seuraaja $\{P_1, \ldots, P_l\}$. Huom! Seuraajia on ainakin yksi.

**Esimerkki.** Solmun

$$C = \{\mathbf{A}(P\mathbf{U}Q), \mathbf{AXA}(P\mathbf{U}Q), \mathbf{EG}P, P, \mathbf{EXEG}P, \mathbf{EF}Q, \mathbf{EXEF}Q\}.$$

seuraajat ovat $D_1 = \{\mathbf{A}(P\mathbf{U}Q), \mathbf{EG}P\}$ ja $D_2 = \{\mathbf{A}(P\mathbf{U}Q), \mathbf{EF}Q\}$.

© 2006 TKK, 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:

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

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## Tulevaisuuslauseiden toteutuminen

- Tulevaisuuslause $\mathbf{EF}Q$ ($\mathbf{E}(P\mathbf{U}Q)$) toteutuu AND-solmussa $C$, joss taulusta löytyy solmusta $C$ lähtevä polku AND-solmuun $C'$, jossa on lause $Q$ (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 seuraavat:
  (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 $P$).

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



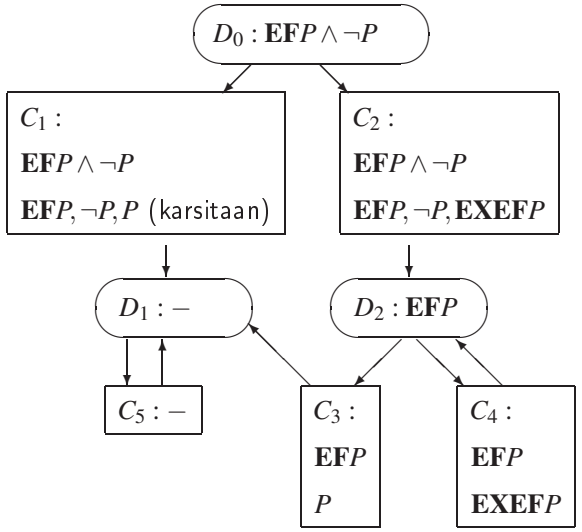
## Esimerkki

- Solmun $D_0 = \{\mathbf{EF}P \wedge \neg P\}$ AND-seuraajat ovat $C_1 = \{\mathbf{EF}P \wedge \neg P, \mathbf{EF}P, \neg P, P\}$ ja $C_2 = \{\mathbf{EF}P \wedge \neg P, \mathbf{EF}P, \neg P, \mathbf{EXEF}P\}$.
- $C_1$:n OR-seuraajat (vähintään yksi): $D_1 = \{\}$.
  $C_2$: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 ja $C_5$:n OR-seuraajat: $\{\} = D_1$.
  $C_4$:n OR-seuraajat: $\{\mathbf{EF}P\} = D_2$.
- Alustava taulu $T_0$ on valmis.
- Karsinta: Poistetaan $C_1$ (lause ja negaatio). Lopullinen taulu $T_1$ valmis (tulevaisuuslause $\mathbf{EF}P$ toteutuu AND-solmuissa $C_2, C_3, C_4$).

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Taulun graafiesitys




© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## 2. Toteutuvuuden ja pätevyyden tutkiminen

- CTL-lauseen $P$ lopullisesta taulusta $T_1$ voidaan muodostaa lauseelle malli, mikäli lause on toteutuva.
- CTL-lauseen $P$ pätevyyttä voidaan tutkia lauseen $\sim P$ toteutumattomuuden kautta.
- LTL-lauseen $P$ toteutuvuus/pätevyys voidaan palauttaa vastaavan CTL-lauseen toteutuvuuteen/pätevyyteen.

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

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## CTL-toteutuvuus tauluilla

CTL-taulu antaa lauseen $P$ mallin:

- Mallin tilat vastaavat AND-solmuja ja valuaatio saadaan suoraan AND-solmun sisältämistä atomilauseista.
- Mallissa on oltava ainakin yksi AND-solmu, johon sisältyy $P$.
- Seuraajat on valittava siten, että malli on sarjallinen ja kaikkien mukaan tulevien AND-solmujen tulevaisuuslauseet toteutuvat.

**Huom.** Taulumenetelmää voidaan käyttää ohjelmasynteesissä (muodostamaan järjestelmän ohjausrakenne):

(i) Annetaan ohjelman määrittely CTL-lauseina.

(ii) Haetaan määrittelylle malli (antaa ohjausrakenteen).

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Esimerkki

Anna CTL-lauseelle $\mathbf{EF}P \wedge \neg P$ voidaan muodosta malli $\langle S, R, v \rangle$ edellä annetusta taulusta $T_1$ esim. seuraavasti:

- Asetetaan $S = \{C_2, C_3, C_5\}$, $R = \{\langle C_2, C_3\rangle, \langle C_3, C_5\rangle, \langle C_5, C_5\rangle\}$ sekä $v(P,s) = \text{true}$, jos $s = C_3$, ja muutoin $v(P,s) = \text{false}$.
- Toinen vaihtoehto: $S = \{C_2, C_3, C_4, C_5\}$, $R = \{\langle C_2, C_4\rangle, \langle C_4, C_3\rangle, \langle C_3, C_5\rangle, \langle C_5, C_5\rangle\}$ sekä $v(P,s) = \text{true}$, jos $s = C_3$, ja muutoin $v(P,s) = \text{false}$.

**Huom.** Esimerkiksi malli, jossa

$S = \{C_2, C_4\}$, $R = \{\langle C_2, C_4\rangle, \langle C_4, C_4\rangle\}$ ja $v(P, C_2) = v(P, C_4) = \text{false}$

ei kelpaa, koska solmujen $C_2$ ja $C_4$ tulevaisuuslause $\mathbf{EF}P$ ei toteudu.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## CTL-pätevyys tauluilla

- Lause $P$ on pätevä, joss sen negaatio $\neg P$ ei ole toteutuva.
- Toteutuvuus voidaan selvittää taulumenetelmällä:
  (i) Muodostetaan lauseen $\neg\varphi$ positiivinen normaalimuoto $\sim\varphi$.
  (ii) Rakennetaan taulu lauseelle $\sim\varphi$.
- Lause $\varphi$ on pätevä, joss $\sim\varphi$ ei ole toteutuva, joss lopullisessa taulussa $T_1$ ei ole yhtään AND-solmua, joka sisältää lauseen $\sim\varphi$.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Esimerkki

Onko CTL-lause

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

pätevä?

Haetaan lauseelle $\neg\varphi$ positiivinen normaalimuoto $\sim\varphi$:

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

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

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

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

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

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## Esimerkki: AND- ja OR-seuraajat

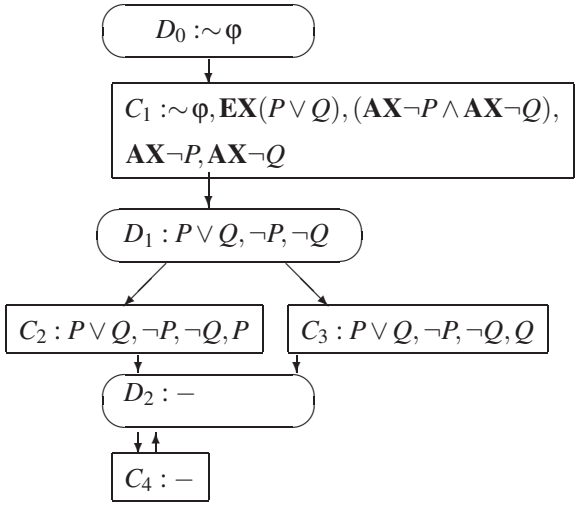
- Solmun $D_0 = \{\sim\varphi\}$ AND-seuraajat:
  $C_1 = \{\sim\varphi, \mathbf{EX}(P \vee Q), (\mathbf{AX}\neg P \wedge \mathbf{AX}\neg Q), \mathbf{AX}\neg P, \mathbf{AX}\neg Q\}$.
- $C_1$:n OR-seuraajat:
  $D_1 = \{P \vee Q, \neg P, \neg Q\}$.
- $D_1$:n AND-seuraajat:
  $C_2 = D_1 \cup \{P\}$ ja $C_3 = D_1 \cup \{Q\}$.
- $C_2$:n OR-seuraajat: $D_3 = \{\}$.
  $C_3$:n OR-seuraajat: $\{\} = D_3$.
  $D_3$:n AND-seuraajat: $C_4 = \{\}$.
  $C_4$:n OR-seuraajat: $\{\} = D_3$.

$\Longrightarrow$ Alustava taulu $T_0$ on valmis.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Esimerkki: alustava taulu




© 2006 TKK, Tietojenkäsittelyteorian laboratorio



## Esimerkki: karsinta

1. Solmut $C_2$ ja $C_3$ voidaan poistaa, koska ne sisältävät lauseen ja sen negaation.
2. OR-solmu $D_1$ voidaan poistaa (kaikki seuraajat poistettu).
3. AND-solmu $C_1$ voidaan poistaa (seuraaja poistettu).
4. OR-solmu $D_0$ voidaan poistaa (kaikki seuraajat poistettu).

$\Longrightarrow$ Lopullinen taulu $T_1$ on valmis.

Taulussa $T_1$ ei ole AND-solmua, jossa esiintyy $\sim\varphi$,
joten $\sim\varphi$ ei ole toteutuva ja $\varphi$ on pätevä.

© 2006 TKK, 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 $\mathbf{F}, \mathbf{G}, \mathbf{X}, \mathbf{U}, \mathbf{B}$ systemaattisesti operaattoreilla $\mathbf{AF}, \mathbf{AG}, \mathbf{AX}, \mathbf{AU}, \mathbf{AB}$.

Tällöin $P$ on LTL-toteutuva joss $P'$ on CTL-toteutuva.

**Esimerkki.** LTL-lause $\mathbf{G}(\neg P \mathbf{U} Q)$ on toteutuva, joss CTL-lause $\mathbf{AGA}(\neg P \mathbf{U} Q)$ on toteutuva.

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## 3. Laskennallinen vaativuus (yhteenveto)

- CTL

  Mallintarkastus: **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

© 2006 TKK, Tietojenkäsittelyteorian laboratorio

## Tehokkaasti toteutettavia aliluokkia

- Toteutuvuusongelma ratkeaa polynomisessa ajassa esim. ohjelmasynteesin kannalta mielenkiintoisissa tapauksissa.
- Esim. SCTL (yksinkertaistettu CTL), jossa pätee ESC-oletus eli tulevaisuuslauseet eivät riipu historiasta:

  $P_1 \vee \cdots \vee P_n$, $\mathbf{AG}(P_1 \vee \cdots \vee P_n)$

  $\mathbf{AG}(P_0 \rightarrow \mathbf{AF}(P_1 \vee \cdots \vee P_n))$,

  $\mathbf{AG}(P_0 \rightarrow \mathbf{A}(P_1 \vee \cdots \vee P_n \mathbf{U} Q_1 \vee \cdots \vee Q_m))$

  $\mathbf{AG}(P_0 \rightarrow \mathbf{AX}(P_1 \vee \cdots \vee P_n) \wedge \mathbf{EX}(Q_1 \vee \cdots \vee Q_m) \wedge \cdots \wedge \mathbf{EX}(R_1 \vee \cdots \vee R_l))$

- Esim. RLTL (rajoitettu LTL):

  $\mathbf{G}(P_1 \vee \cdots \vee P_n)$

  $\mathbf{G}(P_0 \rightarrow \mathbf{F}(P_1 \vee \cdots \vee P_n))$

  $\mathbf{G}(P_0 \rightarrow \mathbf{X}(P_1 \vee \cdots \vee P_n))$

© 2006 TKK, Tietojenkäsittelyteorian laboratorio