

## MALLINTARKASTUS

1. Johdanto mallintarkastukseen
2. Globaali ja lokaali mallintarkastus
3. Toteutustekniikkaa
4. LTL-mallintarkastus
5. Laskennallinen vaativuus (yhteenveto)

E. M. Clarke et al.: *Model Checking*, luku 4 (s. 35–49).

E. A. Emerson: *Automated Temporal Reasoning about Reactive Systems*, luku 3 (s. 16–18).

## Mallin generointi

Mallintarkastusta varten järjestelmän kuvauksesta muodostetaan mahdollisten maailmojen malli  $M$ :

- **Eksplisiittinen esitystapa:**  
Malli  $M$  muodostetaan kuvauksesta saavutettavuusanalyysitekniikalla ennen mallintarkastusta (**tilaräjhdys**).
- **On-the-fly -tekniikka:**  
Malli  $M$  muodostetaan kuvauksesta saavutettavuusanalyysitekniikalla mallintarkastuksen aikana tarpeen mukaan.
- **Symbolinen esitystapa:**  
Tilasiirtymärelaatio esitetään Boolean funktiona symbolisesti.

## 1. Johdanto mallintarkastukseen

Onko annettu lause  $P$  tosi annetussa mallissa  $M$ ?

- Malli  $M$ : järjestelmän malli.  
Saadaan järjestelmän kuvauksesta, joka on usein annettu jollain spesifiointikielillä: SDL, VHDL, prosessialgebra, automaattit, Petri-verkot, SMV, PROMELA ...
- Lause  $P$ : järjestelmän suhteen kiinnostava ominaisuus.  
Annettu usein temporaalilogiikalla: CTL, LTL, CTL\*, ...
  - Täysin automatisoitavissa.
  - Realististen järjestelmien mallit usein suuria.
  - Jo nykyiset tekniikat teollisesti sovellettavissa.

## Tila-avaruuden symbolinen esittäminen

- Järjestelmän kokonaistila esitetään binäärisesti ( $n$  tilabittinä).
- Otetaan käyttöön kullekin tilabitille  $i$  kaksi atomilauseetta  $v_i$  (nykyinen tila) ja  $v'_i$  (uusi tila).
- Määritellään kullekin tilabitille  $i$  tilasiirtymäehdon antava lause  
esim.  $v'_i \leftrightarrow (v_i \wedge v_{i+1}) \vee \neg v_{i+3}$   
joiden konjunktiona saadaan tilasiirtymärelaatiolle lause  $T(\vec{v}, \vec{v}')$ .
- Lause  $T(\vec{v}, \vec{v}')$  kertoo symbolisesti mahdolliset tilasiirtymät:  
Järjestelmä voi siirtyä esim. tilasta  $(0, \dots, 0)$  tilaan  $(1, \dots, 1)$  joss lause  $T(\vec{v}, \vec{v}')$  on tosi mallissa, jossa kaikki  $v_i$  atomit ovat epätosia ja kaikki  $v'_i$  atomit ovat tosia.

### Saavutettavuusrelaation kompositio

- Nyt järjestelmän saavutettaville tiloille voidaan muodostaa lause  $R(\vec{v})$  iteratiivisesti:

1.  $R_0(\vec{v}) := I(\vec{v})$ , missä  $I(\vec{v})$  antaa mahdolliset alkutilat.

2. Toistetaan kaikilla  $i = 1, 2, \dots$

$$R_i(\vec{v}) := \exists \vec{w}(R_{i-1}(\vec{w}) \wedge T(\vec{w}, \vec{v}))$$

kunnes  $R_i(\vec{v}) \equiv R_{i-1}(\vec{v})$  (ovat loogisesti ekvivalentteja).

- $R_i(\vec{v})$  kertoo symbolisesti saavutettavat tilat: esim. tila  $(0, \dots, 0)$  on saavutettavissa joss lause  $R_i(\vec{v})$  on tosi mallissa, jossa kaikki  $v_i$  atomit ovat epätosia.
- Voidaan esittää suuria tila-avaruuksia erittäin tiiviisti: esim.  $R_i(\vec{v}) = v_1$  esittää  $2^{n-1}$  saavutettavaa tilaa (ts. kaikki tilat, joissa tilabitti  $v_1$  on 1).

### Esimerkki: SMV-mallintarkastin

Toteutusteknisiä ratkaisuja:

- Symbolinen tila-avaruuden esitystapa käytössä.
- Tarvittavat lauseet esitetään tehokkaassa OBDD-normaali muodossa (ordered binary decision diagrams).
- Myös temporaalilogiikan lauseen tarkastaminen voidaan tehdä symbolisesti.

### Mallintarkastimet

Mallintarkastin

- ottaa tyypillisesti syötteekseen (i) spesifointikielellä annetun **mallin** ja (ii) temporaalilogiikalla annetun **lauseen** (vaatimusmäärittelyn) ja
- antaa vastauksena ilmoituksen, että lause on tosi mallissa tai **vastaesimerkin** (mallin suorituksen, jossa lause ei toteudu).
- Mallintarkastinta voidaan käyttää siis **“debuggaukseen”** eli virheiden etsimiseen järjestelmän spesifikaatiosta.

### Esimerkki mallin SMV-kuvauksesta

```

MODULE main
    VAR
        state2: {s2, n2};
    VAR
        state1: {s1, n1};
    ASSIGN
        init(state2) := s2;
    ASSIGN
        next(state2) :=
            case
                (state1 = s1) &
                    (state2 = s2): n2;
                (state1 = s1) &
                    (state2 = n2) : {n2, s2};
                (state1 = n1) : {n1, s1};
            1: state2;
    esac;
    SPEC
        AF ((state1 = n1) &
            (state2 = s2))

```

**Esimerkkin mallintarkastuksesta SMVllä**

```

$ smv example.smv
-- specification AF (state1 = n1 & state2 = s2) is false
-- as demonstrated by the following execution sequence
-- loop starts here --
state 1.1:
state1 = s1
state2 = s2

state 1.2:
state1 = n1
state2 = n2

state 1.3:
state1 = s1
state2 = s2

```

© 2007 TKK, Tietojenkäsittelyteorian laboratorio

**Globaali CTL-mallintarkastus**

- Globaali mallintarkastusmenetelmä määrää lauseen totuusarvon mallin kaikissa tiloissa.
  - Tämä tehdään systemaattisesti käsittelemällä vuorollaan lauseen kaikki alilauseet atomilauseista alkaen.
1. Järjestetään lauseen  $P$  alilauseet järjestykseen:

$$P_0, P_1, \dots, P_n (= P),$$

missä kukin  $P_i$  esiintyy vasta kaikkien aitojen alilauseidensa jälkeen.

**Esimerkki.** Eräs alilausejärjestys lauseelle  $\mathbf{A}(P \mathbf{U} \neg P)$ :

$$P, Q, \neg P, \mathbf{E}(Q \mathbf{U} \neg P), \mathbf{A}(P \mathbf{U} \neg P).$$

© 2007 TKK, Tietojenkäsittelyteorian laboratorio

**2. Globaali ja lokaali mallintarkastus**

- Globaali mallintarkastus:  
Missä mallin  $M$  tiloissa annettu lause  $P$  on tosi?
- Lokaali mallintarkastus:  
Onko lause  $P$  tosi mallin  $M$  annetussa tilassa  $s_0$ ?
- Lokaali mallintarkastus (yhdistettynä on-the-fly -tekniikkaan) mahdollistaa mallintarkastuksen, jossa mallin kaikkia tiloja ei tarvitse välttämättä tutkia (eikä edes muodostaa).
- Globaalin mallintarkastuksen toteuttaminen on suoraviivaisempaa ja se voidaan saada tehokkaaksi ja vähemmän muistia käyttäväksi.

© 2007 TKK, Tietojenkäsittelyteorian laboratorio

**CTL-mallintarkastus**

2. Kaikille  $i = 0, 1, \dots, n$  määrätään lauseen  $P_i$  totuusarvo jokaisessa mallin  $M$  tilassa  $s \in S$  seuraavasti:
- (a) Jos  $P_i$  on atomilause, saadaan totuusarvo suoraan mallista.
  - (b) Jos  $P_i$  on muotoa  $\neg P_j$  tai  $P_j \wedge P_l$ , saadaan totuusarvo alilauseiden  $P_j, P_l$  totuusarvoista (Huom! Koska  $j, l < i$ , lauseiden  $P_j, P_l$  totuusarvot on tässä vaiheessa jo määrätty).
  - (c) Jos  $P_i$  on muotoa  $\mathbf{A} \mathbf{X} P_j$ , saadaan totuusarvo alilauseen  $P_j$  totuusarvoista kaikissa  $s$ :n seuraajissa.

**Esimerkki.** Olkoon  $M = (S, R, v)$ , missä  $S = \{s_0, s_1\}$ ,  
 $R = \{\langle s_0, s_0 \rangle, \langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle\}$ ,  $v(s_0, P) = v(s_1, Q) = \text{true}$  ja  
 $v(s_1, P) = v(s_0, Q) = \text{false}$ . Nyt  $M, s_i \models \neg(P \wedge Q)$  kun  $i \in \{0, 1\}$ .

Siis esim.  $M, s_0 \models \mathbf{A} \mathbf{X} \neg(P \wedge Q)$  ja sama voidaan todeta tilasta  $s_1$ .

© 2007 TKK, Tietojenkäsittelyteorian laboratorio

2.(d) Jos  $P_i$  on muotoa  $\mathbf{A}(P_j \mathbf{U} P_l)$ , saadaan sen totuusarvo käyttämällä seuraavaa ekvivalenssia:

$$\mathbf{A}(P_j \mathbf{U} P_l) \equiv P_l \vee (P_j \wedge \mathbf{A}\mathbf{X}\mathbf{A}(P_j \mathbf{U} P_l))$$

i. Merkitään lause  $P_i$  todeksi kaikissa tiloissa, joissa  $P_l$  on tosi.

ii. Merkitään lause  $P_i$  todeksi tilassa  $s$ , jos

$$M, s \models P_j \text{ ja } M, t \models P_i \text{ kaikille } t \text{ joille } sRt,$$

kunnes uusia tällaisia tiloja ei löydy.

iii. Merkitään  $P_i$  epätodeksi muissa tiloissa.

**Esimerkki.** Olkoon  $M = (S, R, v)$ , missä  $S = \{s_0, s_1, s_2, s_3\}$ ,  
 $R = \{\langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle, \langle s_0, s_2 \rangle, \langle s_2, s_3 \rangle, \langle s_3, s_3 \rangle\}$ ,  
 $v(s_i, P) = \text{true}$ , joss  $i \neq 3$ , ja  $v(s_i, Q) = \text{true}$ , joss  $i = 3$ .

Täten  $M, s_3 \models \mathbf{A}(P \mathbf{U} Q)$  kohdan (i) perusteella ja  $M, s_2 \models \mathbf{A}(P \mathbf{U} Q)$  kohdan (ii) perusteella. Muille tiloille  $s_i$  pätee  $M, s_i \not\models \mathbf{A}(P \mathbf{U} Q)$  (iii).

### 3. Toteutustekniikkaa

- Seuraavassa annetaan esimerkki siitä, miten temporaali-operaattoreiden evaluointia voidaan tehostaa niin, että saavutetaan  $\mathcal{O}(|P| * (|S| + |R|))$  aikavaativuus (evaluomalla kukin operaattori ajassa  $\mathcal{O}(|S| + |R|)$ ).

- Käsitellään operaattoreita  $\mathbf{E}(P_j \mathbf{U} P_l)$  ja  $\mathbf{E}\mathbf{G}P_j$ ; huomaa, että

$$\mathbf{A}(P_j \mathbf{U} P_l) \equiv \neg \mathbf{E}(\neg P_l \mathbf{U} (\neg P_j \wedge \neg P_l)) \wedge \neg \mathbf{E}\mathbf{G}\neg P_l.$$

- Muotoa  $\mathbf{E}(P_j \mathbf{U} P_l)$  olevien lauseiden tehokas evaluointi perustuu mallin saavutettavuusrelaation  $R$  hyödyntämiseen taaksepäin.

- Totuusarvon evaluointi on suoritettavissa seuraavaksi esitettävällä CheckEU-algoritmilla ajassa  $\mathcal{O}(|S| + |R|)$ .

2.(e) Jos  $P_i$  on muotoa  $\mathbf{E}(P_j \mathbf{U} P_l)$ , saadaan sen totuusarvo hyödyntämällä seuraavaa ekvivalenssia:

$$\mathbf{E}(P_j \mathbf{U} P_l) \equiv P_l \vee (P_j \wedge \mathbf{E}\mathbf{X}\mathbf{E}(P_j \mathbf{U} P_l))$$

i. Merkitään lause  $P_i$  todeksi kaikissa tiloissa, joissa  $P_l$  on tosi.

ii. Merkitään lause  $P_i$  todeksi tilassa  $s$ , jos

$$M, s \models P_j \text{ ja } M, t \models P_i \text{ jollekin } t \text{ jolle } sRt,$$

kunnes uusia tällaisia tiloja ei löydy.

iii. Merkitään  $P_i$  epätodeksi muissa tiloissa.

- Algoritmin aikavaativuus  $\mathcal{O}(|P| * |S| * (|S| + |R|))$ .
- Temporaalioperaattoreilla alkavien lauseiden evaluointia voidaan parantaa ja päästä aikavaativuuteen  $\mathcal{O}(|P| * (|S| + |R|))$ .
- Globaaliin CTL-mallintarkastusmenetelmään voidaan suoraviivaisesti yhdistää myös reilusehtojen käsittely.

### CheckEU: $\mathbf{E}(P_j \mathbf{U} P_l)$ -lauseiden evaluointi

procedure CheckEU( $P_j, P_l$ )

$T := \{s \mid M, s \models P_l\};$

for all  $s \in T$ , label  $\mathbf{E}(P_j \mathbf{U} P_l)$  true in  $s$ ;

while  $T$  is not empty do

  choose  $s$  in  $T$  and remove it from  $T$ ;

  for all  $t$  such that  $(t, s) \in R$  do

    if  $\mathbf{E}(P_j \mathbf{U} P_l)$  is not yet labeled true in  $t$  and  $M, t \models P_j$  then

      label  $\mathbf{E}(P_j \mathbf{U} P_l)$  true in  $t$ ;

      add  $t$  to  $T$

    endif

  endfor

endwhile

### Vahvasti kytketyt komponentit

- Seuraavaksi esitettävä  $\mathbf{EG}P_j$ -lauseiden tehokas evaluointi perustuu mallin jakamiseen vahvasti kytkettyihin komponentteihin; engl. *strongly connected components* (SCCs).
- Graafin **vahvasti kytketty komponentti**  $C$  on maksimaalinen aligraafi, jossa jokainen solmu on saavutettavissa jokaisesta muusta  $C$ :n solmusta  $C$ :ssä kulkevaa polkua pitkin.
- Komponentti  $C$  on ei-triviaali joss siinä on enemmän kuin yksi solmu tai se sisältää solmun, josta on kaari itseensä.
- Vahvasti kytketyt komponentit voidaan hakea Tarjanin algoritmilla [SIAM J. of Computing, 1(2), 146–160, 1972] lineaarisessa ajassa.

### CheckEG: $\mathbf{EG}P_j$ -lauseiden evaluointi

procedure CheckEG( $P_j$ )

$S' := \{s \in S \mid \mathcal{M}, s \models P_j\}$ ;  $R' := \{(s, t) \in R \mid s, t \in S'\}$ ;

$SCC := \{C \mid C \text{ is a non-trivial SCC of } (S', R')\}$ ;

$T := \{s \mid s \in C \text{ and } C \in SCC\}$  ;

for all  $s \in T$ , label  $\mathbf{EG}P_j$  true in  $s$ ;

while  $T$  is not empty do

  choose  $s$  in  $T$  and remove it from  $T$ ;

  for all  $t$  such that  $t \in S'$  and  $(t, s) \in R'$  do

    if  $\mathbf{EG}P_j$  is not yet labeled true in  $t$  then

      label  $\mathbf{EG}P_j$  true in  $t$ ; add  $t$  to  $T$

    endif

  endfor

endwhile

### $\mathbf{EG}P_j$ -lauseiden evaluointi

Evaluointi perustuu mallin  $\mathcal{M}$  rajoittumaan  $\mathcal{M}' = (S', R', v')$ , joka saadaan mallista  $\mathcal{M}$  poistamalla tilat, joissa  $P_j$  on epätosi:

- $S' = \{s \in S \mid \mathcal{M}, s \models P_j\}$ ,
- $R' = \{(s, t) \in R \mid s, t \in S'\}$  ja
- $v'(s) = v(s)$  kaikille  $s \in S'$ .

Evaluointi perustuu seuraavaan mallien  $\mathcal{M}$  ja  $\mathcal{M}'$  väliseen suhteeseen:

**Lemma.**  $\mathcal{M}, s \models \mathbf{EG}P_j$  joss  $s \in S'$  ja mallissa  $\mathcal{M}'$  löytyy polku tilasta  $s$  tilaan  $t$ , joka on graafin  $(S', R')$  ei-triviaalissa vahvasti kytketyssä komponentissa.

☞ Näin lause  $\mathbf{EG}P_j$  voidaan evaluoida ajassa  $\mathcal{O}(|S| + |R|)$  käyttäen seuraavalla kalvolla annettua CheckEG algoritmia.

### 4. LTL-mallintarkastus

- Seuraavassa esitetään taulujen käyttöön perustuva menetelmä, jolla voidaan tarkastaa, lähtekö annetusta tilasta täysi polku, jossa annettu LTL-lause on tosi.
- Merkitään  $\mathcal{M}, s \models \mathbf{E}P$ , joss on olemassa tilasta  $s$  alkava täysi polku, jossa  $P$  on tosi.
- Tämän menetelmän avulla voidaan vastata myös muihin LTL-mallintarkastuskysymyksiin:  
**Esimerkki.** LTL-lause  $P$  on tosi kaikilla annetusta tilasta  $s$  lähtevillä täysillä poluilla, joss  $\mathcal{M}, s \not\models \mathbf{E}\neg P$ .

**Perusidea**

- Kysymys  $M, s \models EP$  tarkistetaan rakentamalla mallista  $M$  ja lauseesta  $P$  LTL-taulu (Büchi-automaatti), joka kuvaa kaikki mallin tilasta  $s$  lähtevät täydet polut, jotka toteuttavat lauseen  $P$ .
- Taulusta on sitten helppo tarkistaa, onko tällaisia polkuja.
- Muistutus: käsitellään kieltä, jossa konnektiivit ovat  $\neg, \wedge, \mathbf{X}, \mathbf{U}$ . (muut konnektiivit käsitellään lyhennysmerkintöinä: esim.  $P \vee Q = \neg(\neg P \wedge \neg Q)$ ;  $\mathbf{F}P = \top \mathbf{U} P$ ;  $\mathbf{G}P = \neg \mathbf{F} \neg P = \neg(\top \mathbf{U} \neg P)$ ).
- Määritellään mallintarkastusmenetelmää varten apukäsitteet:
  1. Lauseen  $P$  **sulkeuma**  $\text{CL}(P)$ .
  2. **Atomit**  $(s, K)$ , jotka muodostavat LTL-taulun solmut. Näiden avulla voidaan sitten määrittellä varsinaiset LTL-taulut.

**Esimerkki**

Haetaan lauseen  $(\neg H)UC$  sulkeuman  $\text{CL}((\neg H)UC)$  lauseet:

$(\neg H)UC$	$\neg((\neg H)UC)$
$H$	$\neg H$
$C$	$\neg C$
$\mathbf{X}((\neg H)UC)$	$\neg \mathbf{X}((\neg H)UC)$
$\mathbf{X}\neg((\neg H)UC)$	$\neg \mathbf{X}\neg((\neg H)UC)$

(Sulkeuma  $\text{CL}(P)$  on siis lauseen  $P$  laajennettu alilauseiden joukko, jossa on tietty lause ja sen negaatio aina parina mukana).

**Lauseen sulkeuma**

- Lauseen  $P$  **sulkeuma**  $\text{CL}(P)$  on pienin joukko lauseita, joka sisältää lauseen  $P$  ja toteuttaa seuraavat ehdot:
  1.  $\neg P_1 \in \text{CL}(P)$  joss  $P_1 \in \text{CL}(P)$
  2. Jos  $P_1 \wedge P_2 \in \text{CL}(P)$ , niin  $P_1, P_2 \in \text{CL}(P)$ .
  3. Jos  $\mathbf{X}P_1 \in \text{CL}(P)$ , niin  $P_1 \in \text{CL}(P)$
  4. Jos  $\neg \mathbf{X}P_1 \in \text{CL}(P)$ , niin  $\mathbf{X}\neg P_1 \in \text{CL}(P)$
  5. Jos  $P_1 \mathbf{U} P_2 \in \text{CL}(P)$ , niin  $P_1, P_2, \mathbf{X}(P_1 \mathbf{U} P_2) \in \text{CL}(P)$
 (Tässä lause  $\neg \neg Q$  samaistetaan lauseeseen  $Q$ .)
- Intuitionona on, että lauseen  $P$  sulkeuma  $\text{CL}(P)$  on niiden lauseiden joukko, joka voi vaikuttaa lauseen  $P$  totuusarvoon.

**Atomit**

Olkoon annettuna malli  $M = (S, R, v)$  ja tutkittava lause  $P$ .

**Atomi**  $A = (s_A, K_A)$  on pari, missä  $s_A \in S$  ja  $K_A \subseteq \text{CL}(P) \cup \text{AP} \cup \{\top\}$  (AP on kaikkien atomilauseiden joukko) siten, että joukolle  $K_A$  pätee:

1. jokaiselle atomilauseelle  $P \in \text{AP} \cup \{\top\}$ ,  $P \in K_A$ , joss  $M, s_A \models P$ ;
2. jokaiselle  $P_1 \in \text{CL}(P)$ ,  $P_1 \in K_A$ , joss  $\neg P_1 \notin K_A$ ;
3. jokaiselle  $P_1 \wedge P_2 \in \text{CL}(P)$ ,  $P_1 \wedge P_2 \in K_A$ , joss  $P_1 \in K_A$  ja  $P_2 \in K_A$ ;
4. jokaiselle  $\neg \mathbf{X}P_1 \in \text{CL}(P)$ ,  $\neg \mathbf{X}P_1 \in K_A$ , joss  $\mathbf{X}\neg P_1 \in K_A$ ;
5. jokaiselle  $P_1 \mathbf{U} P_2 \in \text{CL}(P)$ ,  $P_1 \mathbf{U} P_2 \in K_A$  joss  $P_2 \in K_A$  tai  $P_1, \mathbf{X}(P_1 \mathbf{U} P_2) \in K_A$ .

**Huom.** Atomeja muodostettaessa lause  $\neg \neg Q$  samaistetaan  $Q$ :hun.

**Atomien muodostamisesta**

Kun halutaan muodostaa kaikki mahdolliset atomit  $(s, K)$ , voidaan käyttää esim. seuraavaa menettelyä:

- Mahdollisten joukkojen  $K$  muodostaminen voidaan nähdä (binäärisenä) hakupuuna (atomitauluna), jonka juuressa ovat tilassa  $s$  todet atomilauseet ja epätosien atomilauseiden negaatiot.
- Puu voi haarautua kullekin lauseelle  $P_1 \in CL(P)$  kahteen haaraan, jossa toisessa on  $P_1$  ja toisessa sen negaatio  $\neg P_1$ .  
(Kussakin joukossa  $K$  jokaisesta lauseesta  $P_1 \in CL(P)$  joko lause  $P_1 \in K$  tai lause  $\neg P_1 \in K$ ).
- Muut säännöt (annettu alla) lisäävät haaraan lauseita, jotka huolehtivat, että atomi muodostuu annettujen ehtojen mukaan.

**Esimerkki**

Olkoon tutkittava lause  $(\neg H)UC$ ,  $AP = \{H, C\}$  ja mallissa  $M$  totuusarvot  $v(s_1, H) = v(s_1, C) = \text{false}$ . Hakupuu (atomitaulu) muodostuu seuraavaksi:

		$\top, \neg H, \neg C$		
	$(\neg H)UC$		$\neg((\neg H)UC)$	
$C$	$\neg H$		$\neg C$	$\neg C$
$\times$	$\mathbf{X}((\neg H)UC)$		$H$	$\neg \mathbf{X}((\neg H)UC)$
	$\neg \mathbf{X}((\neg H)UC)$		$\times$	$\mathbf{X}((\neg H)UC)$

Saadaan mahdollisiksi atomeiksi  $(s_1, K_1)$  ja  $(s_1, K_2)$ , missä  $K_1 = \{\top, \neg H, \neg C, (\neg H)UC, \mathbf{X}((\neg H)UC), \neg \mathbf{X}((\neg H)UC)\}$  ja  $K_2 = \{\top, \neg H, \neg C, \neg((\neg H)UC), \neg \mathbf{X}((\neg H)UC), \mathbf{X}((\neg H)UC)\}$ .

**Säännöt atomitaulujen muodostamiseksi**

$P_1 \in CL(P)$	$P_1 \wedge P_2$	$\neg(P_1 \wedge P_2)$
$P_1 \mid \neg P_1$	$P_1$	$\neg P_1 \mid \neg P_2$
	$P_2$	
$\mathbf{X}P_1$	$\neg \mathbf{X}P_1$	$(P_1 \cup P_2)$
$\neg \mathbf{X}P_1$	$\mathbf{X}P_1$	$P_2$
	$P_1$	$\neg P_2$
	$\mathbf{X}(P_1 \cup P_2)$	$\neg P_1$
	$\neg P_2$	$\neg P_2$
	$\neg \mathbf{X}(P_1 \cup P_2)$	$\neg \mathbf{X}(P_1 \cup P_2)$

- Haara sulkeutuu, jos se sisältää lauseen ja sen negaation.
- Avoin haara  $K$ , joka on valmis (ei uusia lauseita yo. säännöillä ja jokaiselle  $P_1 \in CL(P)$ ,  $P_1 \in K$  tai  $\neg P_1 \in K$ ), on kelvollinen atomin lausejoukoksi  $K$ .

**LTL-taulut**

**Määritelmä.** Mallille  $M = (S, R, v)$  ja lauseelle  $P$  muodostettu **LTL-taulu** on graafi  $G = (N, E)$ , missä

- solmujen joukko  $N$  on mallista  $M$  ja lauseesta  $P$  muodostettavissa olevien atomien  $A$  joukko ja
- kaarien joukko  $E \subseteq N \times N$  pätee:  $(A, B) \in E$ , joss
  1.  $(s_A, s_B) \in R$  ja
  2. jokaiselle  $\mathbf{X}P_1 \in CL(P)$ ,  $\mathbf{X}P_1 \in K_A$ , joss  $P_1 \in K_B$ .

**Esimerkki**

Olkoon tutkittava lause  $(\neg H)UC$  ja malli  $M = (S, R, v)$ , missä

- $S = \{s_1, s_2\}$ ,  $R = \{(s_1, s_2), (s_2, s_2)\}$  ja
- $v(s_1, H) = v(s_1, C) = v(s_2, H) = v(s_2, C) = \text{false}$ .

Tällöin LTL-aulussa  $G = (N, E)$  ovat

$$N = \{ (s_1, K_1), (s_2, K_1), (s_1, K_2), (s_2, K_2) \} \text{ ja}$$

$$E = \{ ((s_1, K_1), (s_2, K_1)), ((s_2, K_1), (s_2, K_1)),$$

$$((s_1, K_2), (s_2, K_2)), ((s_2, K_2), (s_2, K_2)) \}$$

missä joukot  $K_1, K_2$  ovat kuten edellä:

$$K_1 = \{ \top, \neg H, \neg C, (\neg H)UC, \mathbf{X}((\neg H)UC), \neg \mathbf{X}\neg((\neg H)UC) \} \text{ ja}$$

$$K_2 = \{ \top, \neg H, \neg C, \neg((\neg H)UC), \neg \mathbf{X}((\neg H)UC), \mathbf{X}\neg((\neg H)UC) \}.$$

**Itsetoteutuvuus**

**Määritelmä.** LTL-aulun  $G$  vahvasti kytkettyä komponenttia  $C$  sanotaan **itsetoteutuvaksi**, joss jokaiselle atomille  $A \in C$  ja jokaiselle  $P_1UP_2 \in K_A$  on olemassa atomi  $B \in C$  siten, että  $P_2 \in K_B$ .

**Lemma.** LTL-aulussa  $G$  on atomista  $(s, K)$  alkava tulevaisuuspolku, joss atomista  $(s, K)$  on polku johonkin taulun  $G$  itsetoteutuvaan vahvasti kytkettyyn komponenttiin  $C$ .

**Esimerkki.** (Jatkoa) Atomista  $(s_1, K_1)$  alkavaa tulevaisuuspolkua ei ole, koska siitä ei ole polkua itsetoteutuvaan vahvasti kytkettyyn komponenttiin. Huomaa, että  $\{(s_2, K_1)\}$  ei ole itsetoteutuva.

Atomista  $(s_1, K_2)$  alkava tulevaisuuspolku löytyy, koska siitä on polku itsetoteutuvaan vahvasti kytkettyyn komponenttiin  $\{(s_2, K_2)\}$ .

**Tulevaisuuspolut**

**Määritelmä.** **Tulevaisuuspolku** on LTL-aulun  $G$  ääretön polku  $\pi$  siten, että jos  $P_1UP_2 \in K_A$  jollekin polun  $\pi$  atomille  $A$ , niin on olemassa atomi  $B$ , jolle  $P_2 \in K_B$  ja joka on saavutettavissa atomista  $A$  polkua  $\pi$  pitkin.

**Esimerkki.** Polku  $((s_1, K_1), (s_2, K_1), (s_2, K_1), (s_2, K_1), \dots)$  ei ole tulevaisuuspolku, koska  $(\neg H)UC \in K_1$  ja  $C \notin K_1$ . Sen sijaan  $((s_1, K_2), (s_2, K_2), (s_2, K_2), (s_2, K_2), \dots)$  on tulevaisuuspolku.

Tulevaisuuspolut antavat lauseen  $P$  toteuttavia täysiä polkuja:

**Lemma.** Olkoon  $M$  malli,  $P$  lause ja  $G$  vastaava LTL-aulu. Tällöin  $M, s \models \mathbf{E}P$ , joss taulussa  $G$  on tulevaisuuspolku  $\pi$ , joka alkaa atomista  $(s, K)$ , jossa  $P \in K$ .

Tulevaisuuspolkuja voidaan hakea tehokkaasti LTL-aulun **itsetoteutuvien** vahvasti kytkettyjen komponenttien avulla.

**LTL-aulujen ominaisuudet**

**Teoreema.** Olkoon  $M$  malli,  $P$  lause ja  $G$  vastaava LTL-aulu.

Tällöin  $M, s \models \mathbf{E}P$ , joss taulussa  $G$  on atomi  $(s, K)$  siten, että  $P \in K$ , sekä polku atomista  $(s, K)$  johonkin taulun  $G$  itsetoteutuvaan vahvasti kytkettyyn komponenttiin  $C$ .

**Esimerkki.** (Jatkoa) LTL-aulussa  $G$  ei ole atomia  $(s_1, K)$ , jolle pätsi  $(\neg H)UC \in K$  ja josta olisi polku johonkin itsetoteutuvaan vahvasti kytkettyyn komponenttiin.

Näin ollen  $M, s_1 \not\models \mathbf{E}((\neg H)UC)$ .





### LTL mallintarkastusalgoritmi

Edellä annettu teoreema antaa lähtökohdan seuraavalle LTL-mallintarkastusalgoritille, joka on aikavaativuudeltaan  $\mathcal{O}((|S| + |R|) * 2^{\mathcal{O}(|P|)})$ .

Kun halutaan päättää  $M, s \models \mathbf{EP}$ :

1. Muodostetaan LTL-taulu  $G$ .
2. Lasketaan sen vahvasti kytketyt komponentit.
3. Haetaan näistä itsetoteutuvat komponentit  $C$ .
4. Tarkastetaan kaikille atomeille  $(s, K)$ , jolle  $P \in K$ , löytyykö atomista  $(s, K)$  polku johonkin taulun  $G$  itsetoteutuvaan vahvasti kytkettyyn komponenttiin  $C$ .
5. Jos tällainen polku löytyy, niin  $M, s \models \mathbf{EP}$ , muutoin ei.



### 5. Laskennallinen vaativuus (yhteenveto)

- CTL  
Mallintarkastus: **P**-täydellinen  
 $\mathcal{O}(|M| \cdot |P|)$
- LTL  
Mallintarkastus: **PSPACE**-täydellinen  
 $\mathcal{O}(|M| \cdot \exp(|P|))$
- CTL\*  
Mallintarkastus: **PSPACE**-täydellinen  
 $\mathcal{O}(|M| \cdot \exp(|P|))$