Mallintarkastus

Onko annettu lause P tosi annetussa mallissa M?

- Malli M: järjestelmän malli

  Saadaan järjestelmän kuvauksesta, joka on usein annettu jollain spesifiointikielellä: SDL, VHDL, prosessialgebra, automaatit, Petri-verkot, SMV, PROMELA ...

- Lause P: järjestelmän kiinnostava ominaisuus.

  Usein annettu temporaalilogiikalla: CTL, LTL, CTL*, ...

  – Täysin automatisoitavissa.

  – Realististen järjestelmien mallit usein suuria.

  – Jo nykyiset tekniikat teollisesti sovellettavissa.

Mallin generointi

Mallintarkastusta varten järjestelmän kuvauksesta muodostetaan mahdollisten maailmojen malli M.

- Eksplisiittinen esitystapa
  Malli M muodostetaan kuvauksesta saavutettavuusanalyysitekniikalla ennen mallintarkastusta (tilaräjähdys).
- On-the-fly -tekniikka
  Malli M muodostetaan kuvauksesta saavutettavuusanalyysitekniikalla mallintarkastuksen aikana tarpeen mukaan.
- Symbolinen esitystapa
  Tilasiirtymärelaatio esitetään Boolen funktiona symbolisesti.

Esimerkki mallin SMV-kuvauksesta

```
MODULE main

VAR
state1: {s1, n1};
ASSIGN
init(state1) := s1;
next(state1) :=
case
   (state1 = s1) &
          (state2 = s2): n1;
   (state1 = n1) : {n1, s1};
   1: state1;
esac;
```

```
VAR
state2: {s2, n2};
ASSIGN
init(state2) := s2;
next(state2) :=
case
   (state1 = s1) &
          (state2 = s2): n2;
   (state2 = n2) : {n2, s2};
   1: state2;
esac;

SPEC
AF ((state1 = n1) & (state2 = s2))
```


Tila-avaruuden symbolinen esitys (I)

- Järjestelmän kokonaistila esitetään binäärisesti (n tilabittiä).
- Otetaan käyttöön kullekin tilabitille i kaksi atomilausetta v_i (nykyinen tila) ja v_i' (uusi tila).
- Määritellään kullekin tilabitille i tilasiirtymäehdon antava lause

$$v_i' \leftrightarrow (v_i \wedge v_{i+1}) \vee \neg v_{i+3}$$

  joiden konjuntiona saadaan tilasiirtymärelaatiolle lause $T(\vec{v}, \vec{v'})$.
- Lause $T(\vec{v}, \vec{v'})$ kertoo symbolisesti mahdolliset tilasiirtymät: Järjestelmä voi siirtyä esim. tilasta $(0, \ldots, 0)$ tilaan $(1, \ldots, 1)$ joss lause $T(\vec{v}, \vec{v'})$ on tosi mallissa, jossa kaikki v_i atomit ovat epätosia ja kaikki v_i' atomit ovat tosia.

Tila-avaruuden symbolinen esitys (II)

- Nyt järjestelmän saavutettavílle tiloille voidaan muodostaa lause $R(\vec{v})$ iteratiivisesti:
  - $R_0(\vec{v}) := I(\vec{v})$, missä $I(\vec{v})$ antaa mahdolliset alkutilat.
  - Toistetaan kaikilla $i = 1, 2, \ldots$

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

    kunnes $R_i(\vec{v}) \equiv R_{i-1}(\vec{v})$ (ovat loogisesti ekvivalentteja).
- $R_i(\vec{v})$ kertoo symbolisesti saavutettavat tilat: esim. tila $(0, \ldots, 0)$ 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 (eli kaikki tilat, joissa tilabitti v_1 on 1).

Tila-avaruuden symbolinen esitys (III)

- Esimerkiksi SMV-mallintarkastin käyttää symbolista tila-avaruuden esitysmuotoa.
- Tarvittavat lauseet esitetään tehokkaassa OBDD-normaalimuodossa (ordered binary decision diagrams).
- Myös temporaalilogiikan lauseen tarkastaminen voidaan tehdä symbolisesti.

Mallintarkastimet

Mallintarkastin

- ottaa tyypillisesti syötteekseen (i) spesifiointikielellä 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 järjestelmän spesifikaation "debuggaukseen".

Esimerkki: SMV-mallintarkastin

```
> 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
```

Globaali ja lokaali mallintarkastus

- Globaali mallintarkastus:
  Missä mallin tiloissa annettu lause P on tosi?
- Lokaali mallintarkastus:
  Onko lause P tosi mallin 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.

Globaali CTL-mallintarkastus

Globaali mallintarkastusmenetelmä määrää lauseen totuusarvon mallin kaikissa tiloissa.

Tämä tehdään käsittelemällä vuorollaan lauseen kaikki alilauseet alkaen atomilauseista seuraavasti:

(i) Järjestetään lauseen P alilauseet järjestykseen:
$P_0, P_1, \ldots, P_n (= P)$, missä kukin P_i esiintyy vasta kaikkien aitojen alilauseidensa jälkeen.

Esimerkki. Lauseen $\mathbf{A}(P\mathbf{U}\mathbf{E}(Q\mathbf{U}\neg P))$ eräs mahdollinen alilausejärjestys: $P, Q, \neg P, \mathbf{E}(Q\mathbf{U}\neg P), \mathbf{A}(P\mathbf{U}\mathbf{E}(Q\mathbf{U}\neg P))$

CTL-mallintarkastus (II)

(ii) Kaikille $i = 0, 1, \ldots, n$ määrätään lauseen P_i totuusarvo jokaisessa tilassa $s \in S$ seuraavasti:

- Jos P_i atomilause, saadaan totuusarvo suoraan mallista.
- Jos P_i muotoa $\neg P_j$ tai $P_j \wedge P_l$, saadaan totuusarvo alilauseiden P_j, P_l totuusarvoista (Huom. $j, l < i$, joten lauseiden P_j, P_l totuusarvot on tässä vaiheessa jo määrätty).
- Jos P_i muotoa $\mathbf{AX}P_j$, saadaan totuusarvo alilauseen P_j totuusarvoista kaikissa s:n seuraajissa.

CTL-mallintarkastus (III)

- Jos P_i muotoa $\mathbf{A}(P_j\mathbf{U}P_l)$, saadaan sen totuusarvo käyttämällä ekvivalenssia:

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

1. Merkitään lause P_i todeksi kaikissa tiloissa, joissa P_l on tosi.
2. Merkitään lause P_i todeksi tilassa s, jos P_j on tosi siinä ja P_i tosi **kaikissa** sen seuraajissa, kunnes uusia tällaisia tiloja ei löydy.
3. Merkitään P_i epätodeksi muissa tiloissa.

CTL-mallintarkastus (IV)

- Jos P_i muotoa $\mathbf{E}(P_j \mathbf{U} P_l)$, saadaan sen totuusarvo hyödyntämällä ekvivalenssia:

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

  1. Merkitään lause P_i todeksi kaikissa tiloissa, joissa P_l on tosi.
  2. Merkitään lause P_i todeksi tilassa s, jos P_j on tosi siinä ja P_i tosi **jossakin** sen seuraajassa, kunnes uusia tällaisia tiloja ei löydy.
  3. Merkitään P_i epätodeksi muissa tiloissa.

Huom! 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 reiluusehtojen käsittely.

Toteutustekniikkaa

- Seuraavassa annetaan esimerkki siitä, miten temporaalioperaattoreiden evaluointia voidaan tehostaa niin, että saavutetaan $\mathcal{O}(|P| * (|S| + |R|))$ aikavaativuus (evaluoimalla kukin operaattori ajassa $\mathcal{O}(|S| + |R|)$).
- Käsitellään operaattoreita $\mathbf{E}(P_j \mathbf{U} P_l)$ ja $\mathbf{EG} P_j$ ($\mathbf{A}(P_j \mathbf{U} P_l) \equiv \neg \mathbf{E}(\neg P_l \mathbf{U} (\neg P_j \wedge \neg P_l)) \wedge \neg \mathbf{EG} \neg P_l$)
- $\mathbf{E}(P_j \mathbf{U} P_l)$-lauseiden tehokas evaluointi voidaan hoitaa käyttämällä mallin saavutettavuusrelaatiota R taaksepäin.
- Näin $\mathbf{E}(P_j \mathbf{U} P_l)$-lause voidaan evaluoida ajassa $\mathcal{O}(|S| + |R|)$ käyttäen seuraavaa CheckEU-algoritmia.

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

```
procedure CheckEU(P_j, P_l)
    T := {s | M, s ⊨ P_l};
    for all s ∈ T, label E(P_j 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) ∈ R do
            if E(P_j U P_l) is not yet labeled true in t and M, t ⊨ P_j then
                label E(P_j U P_l) true in t;
                add t to T
            endif
        endfor
    endwhile
```


$\mathbf{EG} P_j$-lauseiden evaluointi

- $\mathbf{EG} P_j$-lauseiden tehokas evaluointi perustuu mallin jakamiseen vahvasti kytkettyihin komponentteihin.
- 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.

$\mathbf{EG}P_j$-lauseiden evaluointi (II)

- $\mathbf{EG}P_j$-lauseiden evaluointi perustuu seuraavaan tulokseen, joka koskee mallin $\mathcal{M}$ rajoittumaa $\mathcal{M}' = (S', R', v')$, missä $S' = \{s \in S \mid \mathcal{M}, s \models P_j\}$, $R' = \{(s,t) \in R \mid s,t \in S'\}$ ja $v'(s) = v(s)$ kaikille $s \in S'$ (mallista $\mathcal{M}$ poistetaan kaikki tilat, joissa lause P_j ei ole tosi).

  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.
- Vahvasti kytketyt komponentit voidaan löytää lineaarisessa ajassa $\mathcal{O}(|S'| + |R'|)$ (Tarjanin algoritmi)
- Näin $\mathbf{EG}P_j$-lause voidaan evaluoida ajassa $\mathcal{O}(|S| + |R|)$ käyttäen seuraavaa CheckEG algoritmia.

procedure CheckEG(P_j)
  $S' := \{s \in S \mid \mathcal{M}, s \models P_j\}$; $R' := \{(s,t) \in R \mid s,t \in S'\}$;
  $SCC := \{C \mid C$ is a non-trivial strongly connected component of $(S', R')\}$;
  $T := \{s \mid s \in C, C \in SCC\}$;
  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

LTL-mallintarkastus

- Seuraavassa esitetään taulujen käyttöön pohjaava menetelmä, jolla voidaan tarkastaa, lähteekö annetusta tilasta täysi polku, jossa annettu LTL-lause on tosi.
- Merkitään $M, s \models \mathbf{E}P$ joss on olemassa tilasta s alkava täysi polku, jossa P on tosi.
- Tämän menetelmän avulla voidaan vastata myös muihin LTL-mallintarkastuskysymyksiin.
  Esim. LTL-lause P on tosi kaikilla annetusta tilasta s lähtevillä poluilla joss $M, s \models \mathbf{E}\neg P$ **ei pädе**.

LTL-mallintarkastus

- Perusidea: $M, s \models \mathbf{E}P$ 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 yksinkertaista tarkistaa, löytyykö tällaisia polkuja.
- Muistutus: käsitellään kieltä, jossa konnektiivit ovat $\neg, \wedge, \mathbf{X}, \mathbf{U}$ (muut konnektiivit käsitellään lyhennysmerkintöinä: esim. $P \vee Q = \neg(\neg P \wedge \neg Q)$; $\mathbf{F}P = \top\mathbf{U}P$; $\mathbf{G}P = \neg\mathbf{F}\neg P = \neg(\top\mathbf{U}\neg P)$).
- Seuraavassa määritellään mallintarkastusmenetelmää varten muutama apukäsite:
  - Lauseen P sulkeuma $\mathrm{CL}(P)$
  - Atomit (s, K) (LTL-taulun solmut)

  Näiden avulla voidaan sitten rakentaa LTL-taulut.

Sulkeuma (I)

- Lauseen P sulkeuma $\mathrm{CL}(P)$ on pienin joukko lauseita, joka sisältää lauseen P ja toteuttaa seuraavat ehdot:
  - $\neg P_1 \in \mathrm{CL}(P)$ joss $P_1 \in \mathrm{CL}(P)$
  - Jos $P_1 \wedge P_2 \in \mathrm{CL}(P)$, niin $P_1, P_2 \in \mathrm{CL}(P)$.
  - Jos $\mathbf{X}P_1 \in \mathrm{CL}(P)$, niin $P_1 \in \mathrm{CL}(P)$
  - Jos $\neg\mathbf{X}P_1 \in \mathrm{CL}(P)$, niin $\mathbf{X}\neg P_1 \in \mathrm{CL}(P)$
  - Jos $P_1\mathbf{U}P_2 \in \mathrm{CL}(P)$, niin $P_1, P_2, \mathbf{X}(P_1\mathbf{U}P_2) \in \mathrm{CL}(P)$

  (Tässä lause $\neg\neg Q$ samaistetaan lauseeseen Q.)
- $\mathrm{CL}(P)$ on niiden lauseiden joukko, joka voi vaikuttaa lauseen P totuusarvoon.

Sulkeuma (II)

Esimerkki. Lauseen $(\neg H)\mathbf{U}C$ sulkeuma $\mathrm{CL}((\neg H)\mathbf{U}C)$:

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

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

Atomit

Olkoon annettuna malli $\mathcal{M} = (S, R, v)$ ja tutkittava lause P.
Atomi $A = (s_A, K_A)$ on pari, missä $s_A \in S$ ja $K_A \subseteq \mathrm{CL}(P) \cup \mathrm{AP} \cup \{\top\}$ (AP on kaikkien atomilauseiden joukko) siten, että joukolle K_A pätee:

- jokaiselle atomilauseelle $P \in \mathrm{AP} \cup \{\top\}$, $P \in K_A$ joss $\mathcal{M}, s_A \models P$;
- jokaiselle $P_1 \in \mathrm{CL}(P)$, $P_1 \in K_A$ joss $\neg P_1 \notin K_A$;
- jokaiselle $P_1 \wedge P_2 \in \mathrm{CL}(P)$, $P_1 \wedge P_2 \in K_A$ joss $P_1 \in K_A$ ja $P_2 \in K_A$;
- jokaiselle $\neg\mathbf{X}P_1 \in \mathrm{CL}(P)$, $\neg\mathbf{X}P_1 \in K_A$ joss $\mathbf{X}\neg P_1 \in K_A$;
- jokaiselle $P_1\mathbf{U}P_2 \in \mathrm{CL}(P)$, $P_1\mathbf{U}P_2 \in K_A$ 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 lauseeseen Q.

Atomien muodostaminen (I)

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 $P_1 \in \mathrm{CL}(P)$ kahteen haaraan, jossa toisessa on P_1 ja toisessa sen negaatio $\neg P_1$.
  (Kussakin joukossa K jokaisesta lauseesta $P_1 \in \mathrm{CL}(P)$ joko lause $P_1 \in K$ tai lause $\neg P_1 \in K$).
- Muut säännöt (alla) lisäävät haaraan lauseita, jotka huolehtivat, että atomi muodostuu annettujen ehtojen mukaan.

Atomien muodostaminen (II)

Säännöt atomitaulun rakentamiseen:

$$\frac{P_1 \in \mathrm{CL}(P)}{P_1 \mid \neg P_1} \qquad \frac{P_1 \wedge P_2}{\begin{array}{c}P_1\\P_2\end{array}} \qquad \frac{\neg(P_1 \wedge P_2)}{\neg P_1 \mid \neg P_2}$$

$$\frac{\mathbf{X}P_1}{\neg\mathbf{X}\neg P_1} \qquad \frac{\neg\mathbf{X}P_1}{\mathbf{X}\neg P_1} \qquad \frac{(P_1\mathbf{U}P_2)}{\begin{array}{c|c}P_2 & P_1\\ & \mathbf{X}(P_1\mathbf{U}P_2)\end{array}} \qquad \frac{\neg(P_1\mathbf{U}P_2)}{\begin{array}{c|c}\neg P_2 & \neg P_2\\ \neg P_1 & \neg\mathbf{X}(P_1\mathbf{U}P_2)\end{array}}$$

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

Atomien muodostaminen (III)

Esimerkki. Olkoon tutkittava lause $(\neg H)\mathbf{U}C$, $\mathrm{AP} = \{H, C\}$ ja mallissa $\mathcal{M}$: $v(s_1, H) = v(s_1, C) = \mathrm{false}$. Hakupuu (atomitaulu):

$$\begin{array}{c|c|c}
\top, \neg H, \neg C & & \\
\hline
(\neg H)\mathbf{U}C & & \neg((\neg H)\mathbf{U}C) \\
\hline
C \quad | \quad \neg H & & \neg C \quad | \quad \neg C \\
\times \quad | \quad \mathbf{X}((\neg H)\mathbf{U}C) & & H \quad | \quad \neg\mathbf{X}((\neg H)\mathbf{U}C) \\
\quad\quad | \quad \neg\mathbf{X}\neg((\neg H)\mathbf{U}C) & & \times \quad | \quad \mathbf{X}\neg((\neg H)\mathbf{U}C)
\end{array}$$

Saadaan mahdolliset atomijoukot $(s_1, K_1), (s_1, K_2)$, missä
$K_1 = \{\top, \neg H, \neg C, (\neg H)\mathbf{U}C, \mathbf{X}((\neg H)\mathbf{U}C), \neg\mathbf{X}\neg((\neg H)\mathbf{U}C)\}$
$K_2 = \{\top, \neg H, \neg C, \neg((\neg H)\mathbf{U}C), \neg\mathbf{X}((\neg H)\mathbf{U}C), \mathbf{X}\neg((\neg H)\mathbf{U}C)\}$

LTL-taulut (I)

Mallille $\mathcal{M} = (S, R, v)$ ja lauseelle P LTL-taulu on graafi $G = (N, E)$, missä solmujen joukko N on mallista $\mathcal{M}$ ja lauseesta P muodostettavien atomien joukko ja kaarien joukolle E pätee: $(A, B) \in E$ joss
(i) $(s_A, s_B) \in R$ ja
(ii) jokaiselle $\mathbf{X}P_1 \in \mathrm{CL}(P)$, $\mathbf{X}P_1 \in K_A$ joss $P_1 \in K_B$.

Esimerkki. Olkoon tutkittava lause $(\neg H)\mathbf{U}C$ ja malli $\mathcal{M} = (S, R, v)$ missä $S = \{s_1, s_2\}$, $R = \{(s_1, s_2), (s_2, s_2)\}$ ja
$v(s_1, H) = v(s_1, C) = v(s_2, H) = v(s_2, C) = \mathrm{false}$.

Tällöin LTL-taulussa $G = (N, E)$:
$N = \{(s_1, K_1), (s_2, K_1), (s_1, K_2), (s_2, K_2)\}$ ja
$E = \{\ ((s_1, K_1), (s_2, K_1)), ((s_2, K_1), (s_2, K_1)),$
$\quad ((s_1, K_2), (s_2, K_2)), ((s_2, K_2), (s_2, K_2))\ \}$
missä joukot K_1, K_2 ovat kuten edellisessä esimerkissä.

LTL-taulut (II)

- Tulevaisuuspolku graafissa G on ääretön polku π siten, että jos $P_1\mathbf{U}P_2 \in K_A$ jollakin polun π atomilla A, niin on olemassa atomi B, jossa $P_2 \in K_B$ ja joka on saavutettavissa atomista A polkua π pitkin.

  Esimerkiksi $((s_1, K_1), (s_2, K_1), (s_2, K_1), (s_2, K_1), \ldots)$ ei ole tulevaisuuspolku, koska $(\neg H)\mathbf{U}C \in K_1$ mutta $C \notin K_1$.
  Huom. $((s_1, K_2), (s_2, K_2), (s_2, K_2), (s_2, K_2), \ldots)$ on tulevaisuuspolku.

- Tulevaisuuspolut antavat lauseen P toteuttavia täysiä polkuja.

Lemma. $M, s \models \mathbf{E}P$ joss graafissa G on tulevaisuuspolku π, joka alkaa atomista (s, K), jossa $P \in K$.

LTL-taulut (III)

- Tulevaisuuspolkuja voidaan löytää tehokkaasti vahvasti kytkettyjen komponenttien avulla.
- Graafin G vahvasti kytkettyä komponenttia C sanotaan itsetoteutuvaksi joss jokaiselle atomille $A \in C$ ja jokaiselle $P_1 \mathbf{U} P_2 \in K_A$ on olemassa atomi $B \in C$ siten, että $P_2 \in K_B$.

Lemma. On olemassa atomista (s, K) alkava tulevaisuuspolku joss graafissa G on polku atomista (s, K) johonkin itsetoteutuvaan vahvasti kytkettyyn komponenttiin.

Esimerkki. (Jatkuu) Atomista (s_1, K_1) alkavaa tulevaisuuspolkua ei löydy, koska siitä ei ole polkua itsetoteutuvaan vahvasti kytkettyyn komponenttiin. Huom. $\{(s_2, K_1)\}$ ei ole itsetoteutuva.
Atomista (s_1, K_2) alkava tulevaisuuspolku löytyy, koska siitä löytyy polku itsetoteutuvaan vahvasti kytkettyyn komponenttiin $\{(s_2, K_2)\}$.

LTL-taulut (IV)

Teoreema. $M, s \models \mathbf{E}P$ joss graafissa G on atomi (s, K), jossa $P \in K$ ja löytyy polku atomista (s, K) johonkin itsetoteutuvaan vahvasti kytkettyyn komponenttiin.

Esimerkki. (Jatkuu) Graafissa G ei ole atomia (s_1, K), jossa $(\neg H)\mathbf{U}C \in K$ ja josta löytyy polku johonkin itsetoteutuvaan vahvasti kytkettyyn komponenttiin.

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

LTL mallintarkastusalgoritmi

☞ Teoreema antaa pohjan seuraavalle LTL-mallintarkastusalgoritmille, jonka aikavaativuus on $\mathcal{O}((|S| + |R|) * 2^{\mathcal{O}(|P|)})$.

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

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

Laskennallinen vaativuus

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