



## Mallintarkastus

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

- Malli  $M$ : järjestelmän malli  
Saadaan järjestelmän kuvauksesta, joka on usein annettu jollain spesifiointikielillä: 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.



## 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
EF ((state1 = n1) & (state2 = s2))

```



## Mallin generointi

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

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



## 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:

$P_0, P_1, \dots, P_n (= P)$ , missä kukin  $P_i$  esiintyy vasta kaikkien aitojen alilauseidensa jälkeen.

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



## CTL-mallintarkastus (III)

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

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

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 (II)

(ii) Kaikille  $i = 0, 1, \dots, n$  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  $\mathbf{A}\mathbf{X}P_j$ , saadaan totuusarvo alilauseen  $P_j$  totuusarvoista kaikissa  $s$ :n seuraajissa.



## CTL-mallintarkastus (IV)

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

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

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.



## $\mathbf{EG}P_j$ -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.



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

```



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

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

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

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

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

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

- Perusidea:  $M, s \models \mathbf{EP}$  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 yksinkertaista tarkistaa, löytyykö tällaisia polkuja.
  - Muistutus: käsitellään kieltä, jossa konnektiivit ovat  $\neg, \wedge, \mathbf{X}, \mathbf{U}$  (muut konnektiivit käsitellään lyhennysmerkintöinä: esim.  $P \vee Q = \neg(\neg P \wedge \neg Q)$ ;  $\mathbf{FP} = \top \mathbf{UP}$ ;  $\mathbf{GP} = \neg \mathbf{F}\neg P = \neg(\top \mathbf{U}\neg P)$ ).
  - Seuraavassa määritellään mallintarkastusmenetelmää varten muutama apukäsite:
    - Lauseen  $P$  **sulkeuma**  $CL(P)$
    - **Atomit**  $(s, K)$  (LTL-taulun solmut)
- Näiden avulla voidaan sitten rakentaa LTL-taulut.



## LTL-mallintarkastus

- Seuraavassa esitetään taulujen käyttöön pohjaava menetelmä, jolla voidaan tarkastaa, lähteekö annetusta tilasta täysi polku, jossa annettu LTL-lause on tosi.
- Merkitään  $M, s \models \mathbf{EP}$  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 päde**.



## Sulkeuma (I)

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

## Sulkeuma (II)

**Esimerkki.** Lauseen  $(\neg H)UC$  sulkeuma  $CL((\neg H)UC)$ :

$$(\neg H)UC \quad \neg((\neg H)UC)$$

$$H \quad \neg H$$

$$C \quad \neg C$$

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

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

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

## 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  $P_1 \in CL(P)$  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 (alla) lisäävät haaraan lauseita, jotka huolehtivat, että atomi muodostuu annettujen ehtojen mukaan.

## 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  $K_A \subseteq CL(P) \cup AP \cup \{\top\}$  ( $AP$  on kaikkien atomilauseiden joukko) siten, että joukolle  $K_A$  pätee:

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

Säännöt atomitaulun rakentamiseen:

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

- 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 joukko  $K$ .



## Atomien muodostaminen (III)

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

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

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



## LTL-taulut (II)

- **Tulevaisuuspolku** graafissa  $G$  on ääretön polku  $\pi$  siten, että jos  $P_1UP_2 \in K_A$  jollakin polun  $\pi$  atomilla  $A$ , niin on olemassa atomi  $B$ , jossa  $P_2 \in K_B$  ja joka on saavutettavissa atomista  $A$  polkua  $\pi$  pitkin.

Esimerkiksi  $((s_1, K_1), (s_2, K_1), (s_2, K_1), (s_2, K_1), \dots)$  ei ole tulevaisuuspolku, koska  $(\neg H)UC \in K_1$  mutta  $C \notin K_1$ .

Huom.  $((s_1, K_2), (s_2, K_2), (s_2, K_2), (s_2, K_2), \dots)$  on tulevaisuuspolku.

- Tulevaisuuspolut antavat lauseen  $P$  toteuttavia täysiä polkuja.

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



## 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  $\mathbf{X}P_1 \in CL(P)$ ,  $\mathbf{X}P_1 \in K_A$  joss  $P_1 \in K_B$ .

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

Tällöin LTL-taulussa  $G = (N, E)$ :

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

missä joukot  $K_1, K_2$  ovat kuten edellisessä esimerkissä.



## LTL-taulut (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  $P_1UP_2 \in K_A$  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  $(\neg H)UC \in K$  ja josta löytyy polku johonkin itsetoteutuvaan vahvasti kytkettyyn komponenttiin.

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



## Laskennallinen vaativuus

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



## 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.