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 ...
- 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 (tiloräjähdys).
- On-the-fly -tekniikka
  Malli M muodostetaan kuvauksesta saavutettavuusanalyysitekniikalla mallintarkastuksen aikana tarpeen mukaan.
- Symbolinen esitystapa
  Tilasiirtymärelaatio esitetään Boolen 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: $P_0, P_1, \ldots, P_n (= P)$, 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: $P, Q, \neg P, \mathbf{E}(Q\mathbf{U}\neg P), \mathbf{A}(P\mathbf{U}\mathbf{E}(Q\mathbf{U}\neg P))$

(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.

- 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.

- 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 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)
  $T := \{s \mid M, s \models P_l\}$;
  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 s is reachable from t 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

- $\mathbf{EG} P_j$ 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 perustuu seuraavaan tulokseen, joka koskee mallin $\mathcal{M}$ rajoittumaa $\mathcal{M}' = (S', R', v')$, joka saadaan mallista $\mathcal{M}$ poistamalla siitä 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)
  $S' := \{s \mid M, s \models P_j\}$;
  $SCC := \{C \mid C$ 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 s is reachable from t 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 päde**.

LTL-mallintarkastus

- 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 muutama apukäsite:
  - Lauseen P sulkeuma $\mathrm{CL}(P)$, joka on niiden lauseiden joukko, joka voi vaikuttaa lauseen P totuusarvoon.
  - Atomit (s, K), jotka antavat LTL-taulun mahdolliset solmut

  Näiden avulla voidaan sitten rakentaa LTL-taulut.

Sulkeuma

- 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.

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 $P_1, \mathbf{X}(P_1 \mathbf{U} P_2) \in K_A$;

Atomien muodostaminen

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

- Mahdollisten joukkojen K muodostaminen voidaan nähdä (binäärisenä) hakupuuna (tauluna), 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 lauseen $P_1 \in \mathrm{CL}(P)$ joko lause $P_1 \in K$ tai lause $\neg P_1 \in K$).
- Haaraan voidaan lisätä lauseita atomille annettujen ehtojen mukaan.
- Haara sulkeutuu, jos se sisältää lauseen ja sen negaation.
- Valmis (atomin ehdot täyttävä), avoin haara K (jossa siis jokaiselle $P_1 \in \mathrm{CL}(P)$ joko $P_1 \in K$ tai $\neg P_1 \in K$) on kelvollinen joukko K.

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 (taulu):

$$\top$$
$$\neg H$$
$$\neg C$$

Vasen haara: $(\neg H)\mathbf{U}C$

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

Oikea haara: $\neg((\neg H)\mathbf{U}C)$

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

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

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

- Rakennetaan taulu (graafi) $G = (N, E)$, missä solmujen joukko N on 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 $P_1 \in K_B$.
- 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.
- 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 $P \in K$.

- 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 kytketty komponenttiin.

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.

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

LTL mallintarkastusalgoritmi

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

1. Muodostetaan 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.

Laskennallinen vaativuus

- CTL

  Mallitarkastus: **P**-täydellinen

  $O(|M| \cdot |P|)$
- LTL

  Mallitarkastus: **PSPACE**-täydellinen

  $O(|M| \cdot exp(|P|))$
- CTL*

  Mallitarkastus: **PSPACE**-täydellinen

  $O(|M| \cdot exp(|P|))$