MARIA: Saavutettavuusanalysaattori korkean tason verkoille

Marko Mäkelä
Tietojenkäsittelyteorian laboratorio
PL 9700, 02015 TKK

29. lokakuuta 2001

Rinnakkaisten järjestelmien analyysivälineitä

Järjestelmissä, joissa esiintyy rinnakkaisuutta, voi olla virheitä, jotka tuntuvat esiintyvän täysin satunnaisesti ja ovat erittäin hankalasti toistettavissa. Tällaiset virheet voivat tulla kalliiksi hajautetuissa järjestelmissä, joissa asiakkailla on päätelaitteita suuria määriä.

Kaikkia rinnakkaisuuteen liittyviä virheitä ei voi löytää debuggerilla eikä testaamalla. Se, että testeissä ei havaita virheitä, ei takaa järjestelmää virheettömäksi. Ainoastaan, jos analyysi kattaa järjestelmän kaikki mahdolliset tilat ja suoritukset, voidaan olla varmoja virheettömyydestä eli siitä, että järjestelmä (tai sitä toivottavasti oikein kuvaava malli) noudattaa vaadittuja ominaisuuksia (jotka on toivottavasti valittu järkevästi).

Eräs helposti automatisoitava menetelmä on *saavutettavuusanalyysi* eli kaikkien suoritusten tai tilojen tutkiminen.

Saavutettavuusanalysaattorit Teknillisessä korkeakoulussa

1984–1988: PRENA (tuhansia saavutettavia tiloja); Pr/T-verkot Pascal-kielisin lisäyksin

1989–: PROD (miljoonia saavutettavia tiloja); Pr/T-verkot C-kielisin lisäyksin

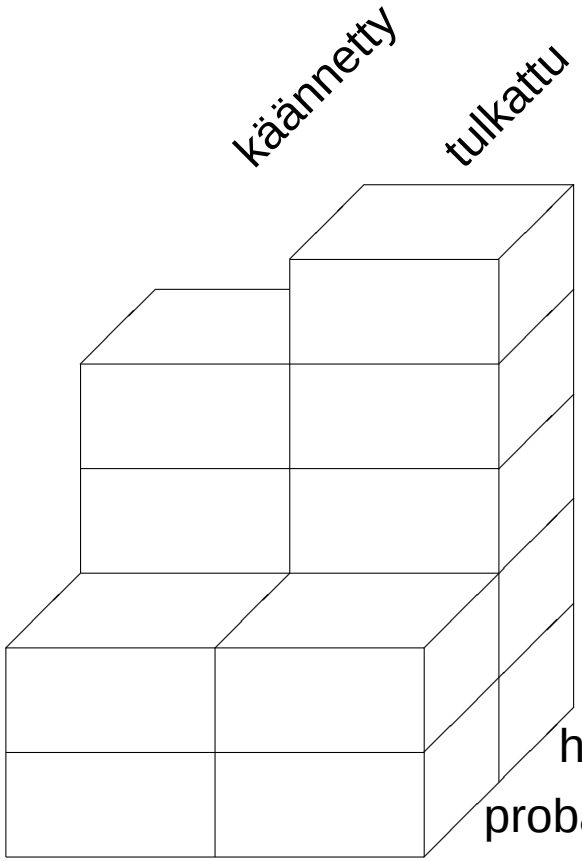
- lennosta verifiointi, itsepäiset joukot, . . .

1998–: MARIA (kymmeniä miljoonia saavutettavia tiloja); algebralliset järjestelmäverkot

- kaksi toimintatilaa: tulkattu (C++) ja käännetty (C/C++)
- modulaarinen rakenne: helppo toteuttaa uusia algoritmeja

Työkaluilla on tutkittu mm. kahta VR:n PLC-pohjaista kulunvalvontajärjestelmää, VTT:n ATM-kytkimen FSR-väylää, ETSI:n ISDN-protokollaa DSS.1 ja 3G:n RLC-protokollaa.

MARIAn ominaisuuksia



käännös matalan tason verkoksi

vuorovaikutteinen simulaatio

elävyystarkistukset lennosta

turvallisuustarkistukset lennosta

kattava saavutettavuusanalyysi

häviötön haku

probabilistinen haku

Eri toimintatilat

MARIA voi käsitellä järjestelmämalleja kahdella eri tavalla. Se joko tulkitsee malleja tai kääntää mallin joukoksi C-kielisiä moduleja, jotka linkitetään konekieliseksi kirjastoksi.

Tulkkitila on hyödyllinen muokattaessa ja simuloitaessa mallia. Seuraajatilojen laskenta kestää kauemmin kuin käännettyä mallia käytettäessä, mutta kääntämisen aiheuttama aloitusviive poistuu.

Kääntäjätilasta on eniten apua tutkittaessa tila-avaruuksia eräajona.

MARIAn graafialgoritmit

MARIAssa on joitakin perustyökaluja saavutettavuusgraafien tutkimiseen:

- tilan edeltäjä ja seuraajatilojen luetteleminen,
- kahden tilan tai tilan ja tilajoukon välisen lyhimmän polun määrittäminen,
- vahvasti kytkettyjen komponenttien muodostaman graafin tutkiminen ja
- halutun ominaisuuden vastaisen suorituspolun esittäminen.

Polkuja ja graafeja on mahdollista nähdä sekä tekstinä että kuvina.

Liitännät muihin työkaluihin (1/3)

MARIA voidaan liittää joihinkin muihin työkaluihin. Osa työkaluista toimii aliohjelman tavoin:

- C-kääntäjä ja linkkeri mallien kääntämiseen,
- erillinen työkalu LTL-aikalogiikan kaavojen kääntämiseksi yleistetyiksi Büchi-automaateiksi ja
- AT&T-laboratorion GraphViz tila-avaruuksien esittämiseen.

Liitännät muihin työkaluihin (2/3)

Tutkimusryhmässämme on kehitetty työkaluja MARIA-mallien tuottamiseksi muista määrittelykielistä:

- SDL, CCITT Specification and Description Language (kehitteillä)
- TNSDL, TeleNokia SDL (perustuu TNSDL–PROD-kääntäjään EMMAan)
- TTKK:n TVT-työkalun tila-transitiojärjestelmät

Lisäksi Brandenburgin teknillisessä korkeakoulussa Cottbusissa Saksassa analysoidaan ohjelmoitavia logiikkaohjaimia (PLC) MARIAlla.

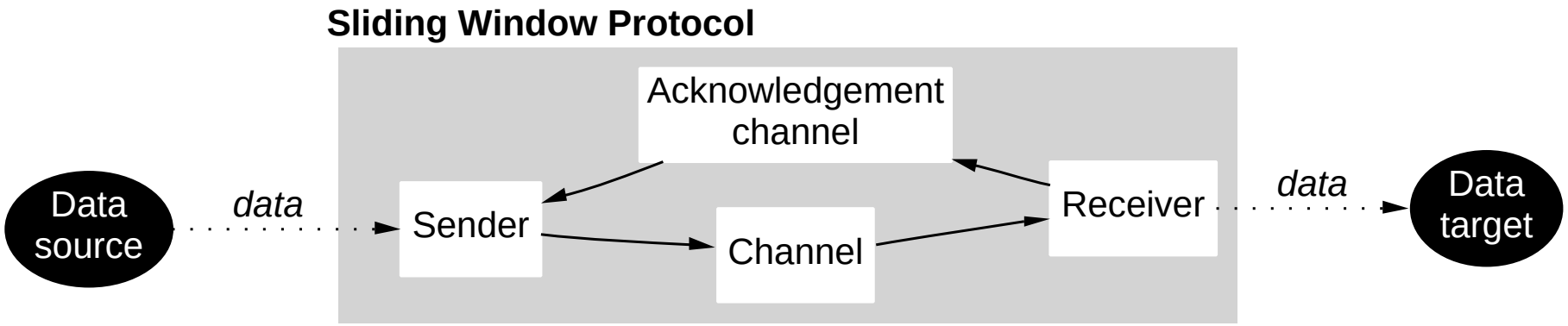
Liitännät muihin työkaluihin (3/3)

MARIA-malleja voidaan muuntaa kahteen perustavasti erilaiseen muotoon.

Malli voidaan keriä auki matalan tason Petri-verkoksi LoLAn, PEPin tai PRODin ymmärtämässä muodossa. Nämä työkalut osaavat tällä hetkellä kutistaa tila-avaruuksia tehokkaammin kuin MARIA.

Mallin tila-avaruus tai sen osa on mahdollista tallentaa GraphViz-ohjelmiston ymmärtämänä suunnattuna graafina tai TVT:n tila-transitiojärjestelmänä.

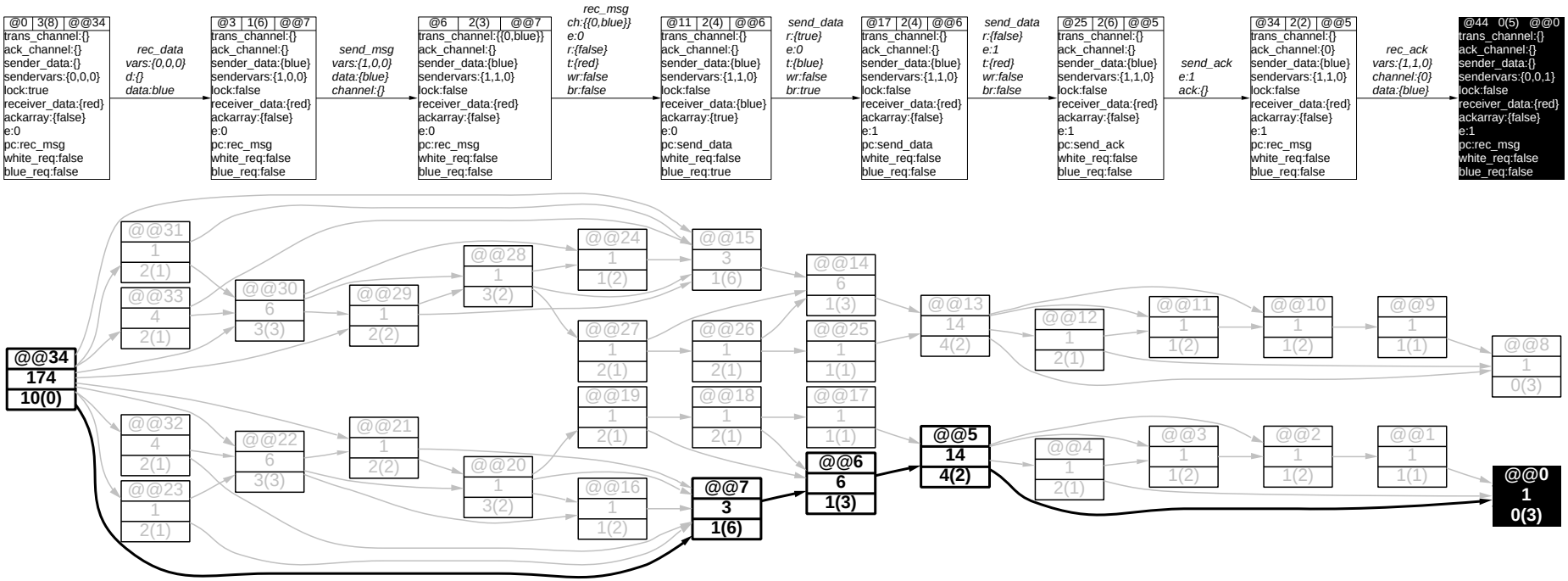
Esimerkki: liukuvan ikkunan protokollan verifiointi (1/3)



Olemme tehneet protokollasta mallin työkalumme kielellä (reilut 200 riviä tekstiä). Tässä mallin versiossa lähetetään kahdenlaisia viestejä, kunnes kolmannenlainen viesti katkaisee lähetyksen. Sekä lähettäjän että vastaanottajan puskurin koko on 1 viesti. Mallin saavutettavuusgraafissa on 264 tilaa ja 582 kaarta. Jos liikenne ei katkeaisi, mallin jokaisesta tilasta pääsisi takaisin alkutilaan (ja edelleen jokaiseen tilaan) eli saavutettavuusgraafilla olisi yksi *vahvasti yhtenäinen komponentti*.

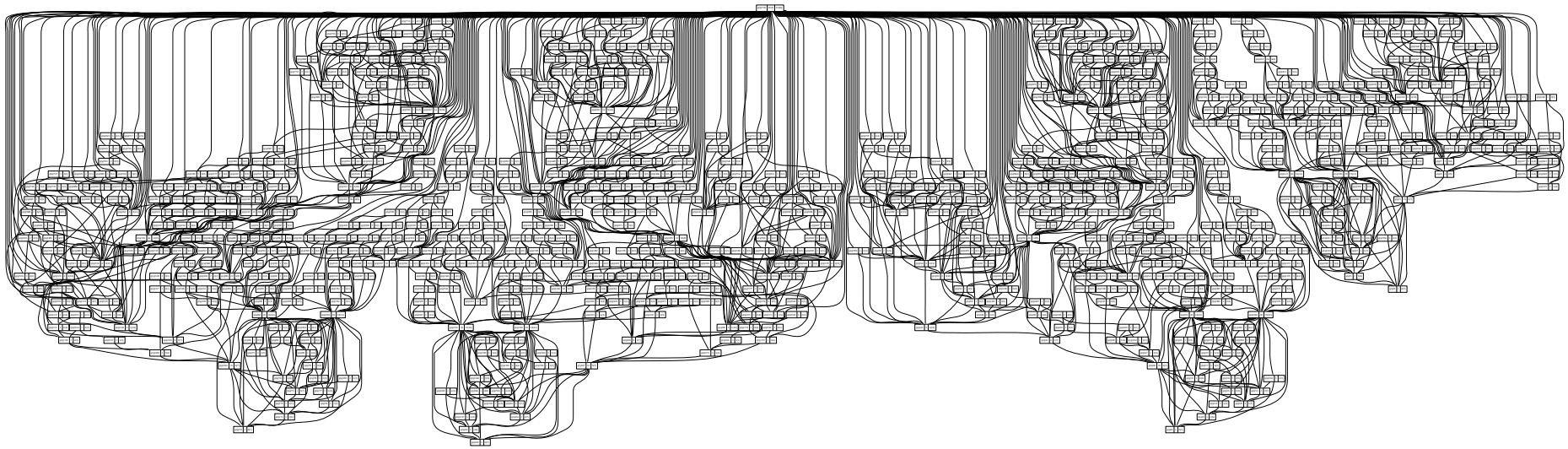
Esimerkki: liukuvan ikkunan protokollan verifiointi (2/3)

Koska malli lukkiutuu katkaisevan viestin lähetyksen jälkeen, sen saavutettavuusgraafissa on useita vahvasti yhtenäisiä komponentteja:



Esimerkki: liukuvan ikkunan protokollan verifiointi (3/3)

Kun jonojen enimmäispituutta kasvatetaan, järjestelmän tila-avaruus kasvaa räjähdysmäisesti. Ikkunakoolla 2 tiloja syntyy 8 268 ja siirtymiä 25 168. Komponentteja on 737 ja niiden välisiä siirtymiä 2 048.



MARIAn suorituskyky

MARIA kykenee käsittelemään huomattavasti suurempia tila-avaruuksia kuin on järkevää esittää yhdellä kuvalla. Ei ole mitään tarvetta esittää kaikkia järjestelmän tiloja, jos olemme kiinnostuneita vain jostakin suorituspolusta, josta ilmenee virheellistä käyttäytymistä. Koska MARIA toimii myös vuorovaikutteisena työkaluna, tulosten analysointi on helppoa.

Suurin tähän mennessä analysoitu järjestelmä on erään radioliikenneprotokollan yksinkertaistettu malli. Sen saavutettavuusgraafissa on 15 866 988 tilaa ja 61 156 129 siirtymää. Analyysi vei 10 tuntia 266 MHz:n Pentium II:n laskenta-aikaa ja 1,55 GB levytilaa mutta vain 5 MB muistia. (Silloin ei vielä ollut käännösvalitsinta, joka mahdollistaa saavutettavuusgraafitiedostojen käsittelyn suoraan muistin kautta.)

Levytilasta suurin osa kului siirtymien muuttujien esittämiseen. Vaikka mallissa on käytetty monimutkaisia rakenteisia tietotyyppejä, MARIA käytti 40–50 tavua kutakin mallin tilaa ja reilut 10 tavua kaarta kohden.

Tutkittavat ominaisuudet (1/2)

Täysin automaattisesti rinnakkaisista järjestelmistä voidaan löytää karkeita virheitä kuten

- lukkiumia eli sellaisia tiloja, joista järjestelmä ei voi edetä, ja
- virheellisiä laskenta-askelia, joissa tapahtuu jokin virhe.

Järjestelmää tuntevat tahot voivat kuvata siltä edellyttämiään *turvallisuus*- ja *elävyysominaisuuksia* esimerkiksi aika- tai modaalilogiikan kaavojen avulla.

Jos analysaattori havaitsee järjestelmän käyttäytyvän haluttujen ominaisuuksien vastaisesti, se tulostaa *vastaesimerkin* eli jonkin ominaisuuden vastaisen tapahtumaketjun, joka järjestelmän on mahdollista suorittaa.

Tutkittavat ominaisuudet (2/2)

Turvallisuusominaisuuden (“mitään pahaa ei tapahdu”) vastainen suoritus on tapahtumaketju järjestelmän alkutilasta johonkin järjestelmässä mahdolliseen “pahaan” tilaan.

Elävyysominaisuus (“jotain hyvää tapahtuu joskus”) rikkoutuu, jos järjestelmä voi loputtomasti suorittaa jotakin tapahtumasilmukkaa suorittamatta “hyvää” toimintoa lainkaan. Se on eräässä mielessä verrattavissa suorituskykyvaatimukseen: hyötylaskennan on edettävä jossakin äärellisessä ajassa. Tarkkaa aikaa formalismimme ei kykene ilmaisemaan.

Turvallisuusominaisuuksia on huomattavasti helpompi tarkistaa kuin elävyysominaisuuksia. Yleensä tällainen tarkistus yhdistetään *probabilistiseen verifiointiin*, jonka yksittäisessä ajossa katetaan koko tila-avaruus jollakin todennäköisyydellä. Toistamalla ajoja voidaan pienentää sitä todennäköisyyttä, että mahdollisia tapahtumaketjuja jää tutkimatta.

Sovellettavuus (1/2)

MARIAn kehittämisessä perimmäinen ajatus oli luoda saavutettavuusanalysaattori ja mallintarkistin sellaiselle mallinnuskielelle, joka on ilmaisuvoimaltaan lähellä korkean tason ohjelmointi- ja spesifiointikieliä (kuten C++ ja SDL). Kielen tehokkaat operaatiot mahdollistavat monimutkaisenkin kommunikaation kuvaamisen lisäämättä malliin turhia välitiloja.

Käyttäjien ei tarvitse tuntea analysaattorimme formalismia. He käyttävät oman sovellusalueensa kieltä, ja *sovellusalakohtainen etupää* vastaa liitännästä analysaattoriin

- kääntämällä käyttäjän ohjelmat tai spesifikaatiot analysaattorin omaan formalismiin,
- mahdollistamalla haluttujen ominaisuuksien kuvaamisen sovelluksen kielellä ja
- esittämällä löydetyt virheelliset käyttäytymiset sovelluksen suorituskaaviona.

Sovellettavuus (2/2)

Yleisellä formalismilla on joitakin etuja sovelluskohtaisiin formalismeihin nähden. Tehokkaampien analyysimenetelmien toteuttaminen hyödyttää heti kaikkia niitä kieliä, joiden kääntäminen analysaattorin kielelle on automatisoitu.

Tällä hetkellä meillä on kokeelliset etupäät puhelinalan järjestön ITU-T:n standardoimalle SDL-kielelle sekä Nokian kehittämälle TNSDL-kielelle. MARIAn tyyppijärjestelmän ansiosta lausekkeiden ja sanomanvälityksen kääntäminen on helppoa.

Kehitämme Java-kielen etupäätä yhdysvaltalaisen Kansasin yliopiston SAnToS-laboratorion Bandera-ohjelmiston pohjalta. Ohjelmisto perustuu kanadalaisen McGill-yliopiston Sable-ryhmän kehittämään SOOT-kalustoon. On kiinnostavaa vertailla tuloksia muihin Banderan tukemiin analysaattoreihin.

MARIA soveltuu erinomaisesti myös järjestelmien mallintamiseen käsin.

Saatavuus

MARIA on vapaasti kenen tahansa hyödynnettävissä samoin ehdoin kuin esimerkiksi Linux-käyttöjärjestelmä, sillä sen lisenssinä on GNU-hankkeen julkaisema General Public License. Uskomme, että tällä tavalla kynnys työkaluun tutustumiseen ja sen laajentamiseen madaltuu.

Analysaattori on kehitetty UNIX-ympäristössä. Osa siitä toimii missä tahansa järjestelmässä, jolle on saatavissa standardin mukainen C++-kääntäjä.

Jos kiinnostuksesi heräsi, tutustu kotisivuihimme osoitteessa

`http://www.tcs.hut.fi/maria/`

ja ota meihin yhteyttä. Etsimme jatkuvasti kiinnostavia järjestelmiä, joiden mallintaminen ja analysointi auttaa kehittämään työkaluamme.