

EXAMENSARBETE Improved precision and verification for test selection in Modelica**STUDENT** Markus Olsson, Filip Stenström**HANDLEDARE** Niklas Fors (LTH), Jon Sten (Modelon AB)**EXAMINATOR** Görel Hedin (LTH)

Förbättrat testurval för Modelica

POPULÄRVETENSKAPLIG SAMMANFATTNING **Markus Olsson, Filip Stenström**

Testurvalstekniker för mjukvara kan spara mycket tid vid testning. I detta arbete har en ny testurvalsteknik för modelleringspråket Modelica definierats, med högre precision än tidigare tekniker och med verifierad säkerhet.

I dagens mjukvaruindustri släpps det kontinuerligt nya uppdateringar. För att säkerställa att dessa uppdateringarna inte inför nya buggar, kan man använda så kallade regressionstester. Det innebär att man testar programmets funktionalitet för att upptäcka om något oförutsett ändras. Ett problem är att det kan ta lång tid att köra alla tester. Men det är onödigt att köra tester som inte har påverkats av en ändring. Därför kan man använda en testurvalsteknik för att välja ut och köra de tester som man tror kan ha påverkats av uppdateringen. Om urvalstekniken garanterat väljer alla tester som har påverkats så kallas den *säker*.

I vårt examensarbete har vi definierat en säker testurvalsteknik för modelleringspråket Modelica. Vårt mål var att förbättra precisionen – det vill säga att minska antalet oförändrade tester som valdes ut – jämfört med en tidigare teknik och det lyckades vi med. Den genomsnittliga testtiden minskade med ca 96% för Modelicas standardbibliotek vid små ändringar jämfört med att köra alla tester. Vår teknik minskar testkörtiden med ca 35% jämfört med den enda tidigare testurvalstekniken för Modelica som vi känner till.

För att utvecklare ska våga använda en testurvalsteknik är det viktigt att visa att den är säker. I vårt arbete analyserade vi först Modelicas standardbibliotek för att hitta beroenden från tester till vanliga klasser. Med hjälp av denna information kunde vi sen skapa en uppsättning tester

som verifierar att en urvalsteknik för Modelica är säker. En intressant upptäckt under vår verifiering av teknikens säkerhet var att den tidigare urvalstekniken faktiskt inte var säker. För att göra den säker lade vi därför till det som saknades i dess implementation. Det var den säkra versionen som vi använde till tidsmätningarna.

Vår teknik är baserad på att analysera beroenden mellan Modelica-klasser. En stor del av vårt examensarbete var att definiera reglerna för dessa klassberoenden. Tekniken fungerar på så sätt att den först hittar alla beroenden för testklasserna i ett Modelica-projekt, och väljer sedan ut de tester som har beroenden på klasser som användaren har ändrat på.

För att tekniken ska spara tid är det viktigt att beroendeanalysen inte tar längre tid än vad det hade tagit att köra alla testerna. För t.ex. Java-program kan detta vara ett problem, men i Modelica tar testerna så lång tid att vår analysid är under 0,3% av den totala testtiden.

En del av verifieringen var att utföra automatiska ändringar i varje klass, en i taget, och undersöka vilka tester som påverkas. Om ett test påverkades, så innebär det att testklassen beror på den ändrade klassen. Om beroendeanalysen inte hittar det beroendet så är testurvalstekniken inte säker.