

EXAMENSARBETE Type Checker Generation using Reference Attribute Grammars**STUDENT** Nicholas Boyd Isacson**HANDLEDARE** Christoph Reichenback (LTH)**EXAMINATOR** Niklas Fors (LTH)

Autogenererad typsäkerhet för programmeringsspråk

POPULÄRVETENSKAPLIG SAMMANFATTNING **Nicholas Boyd Isacson**

Typkontrollerare är ett vanligt verktyg för att hitta buggar i datorprogram, men är relativt komplicerade att skapa. Jag har byggt ett program som automatiskt skapar typkontrollerare för program utifrån en kort definition.

Datorprogram hanterar ofta många olika datatyper, och fel användning av dem är en av de vanligaste buggarna. För att hitta dessa buggar används typkontrollerare, som systematisk går igenom hela programmet och verifierar varenda användning av variabler eller värden för att se till att rätt datatyp använts på rätt plats.

```
Type mismatch: cannot convert from int to boolean Java(16777233)
```

[View Problem \(Alt+F8\)](#) No quick fixes available

```
if (0) {  
    System.err.println("Hello!");  
}
```

Ett typfel i ett vanligt programmeringsspråk, där en siffra används istället för ett booleskt (sant/falskt) värde.

Dessa typkontrollerare är dock komplicerade att skapa, och är ofta starkt kopplade till ett specifikt programmeringsspråk. Det innebär att varenda programmeringsspråk behöver skapa sin egen typkontrollerare, och leder till en otrolig mängd upprepat arbete. I mitt arbete har jag skapat ett verktyg för att generera typkontrollerare per automatik, utifrån en kort och koncis definition av

programmets regler.

Detta görs genom att baka in kod i ett av mellanleden när programmet omvandlas från mänskligskrivna text till maskinexekverbar kod. Mellanledet har en struktur lik ett släktträd där allt härstammar ifrån roten, med grenar som representerar olika kodstrukturer, och som i sig kan delas av i flera grenar.

Därmed kan vi enkelt skapa typkontrollerare för nya språk på ett kortfattat vis, genom att bara definiera vilka regler som ska finnas och inte behöva fundera över hur man implementerar dem. Det underlättar också om man vill göra förändringar i systemet, eftersom det är mycket lättare att ändra de koncisa reglerna än att ändra de många raderna kod som kan krävas för att implementera den.

I nuläget är programmet ganska begränsat i hur komplicerade typkontrollerare det kan generera, och skulle inte kunna appliceras på något av de stora, kända programmeringsspråken. Däremot klarar det flera exempel från forskningslitteraturen, och tillvägagångssättet har dock visats vara flexibelt och utbyggbart. Man borde alltså kunna vidareutveckla programmet för ett bredare stöd i framtiden.