

De onderzoeksgroep
Software Languages Lab

nodigt U graag uit op de openbare verdediging van het proefschrift van

Noah Van Es

ter behaling van de graad van Doctor in de wetenschappen

Titel van het proefschrift:

**Abstract Garbage Collection
for Efficient and Precise Abstract Interpretation**

Curriculum vitae

Promotor:
Prof. dr. Coen De Roover (VUB)

De verdediging heeft plaats op
vrijdag 23 oktober 2026 om 16u

Campus Etterbeek VUB, Pleinlaan 2, Elsene,
Promotiezaal D.2.01

De verdediging is ook te volgen via een
livestream. Een link kan op voorhand worden
aangevraagd via noah.van.es@vub.be.

Samenstelling van de jury

Prof. dr. Wolfgang De Meuter (VUB, voorzitter)
Prof. dr. Jens Nicolay (VUB)
Prof. dr. Joost Vennekens (VUB)
Dr. Caterina Urban (INRIA Paris, FR)
Dr. Raphaël Monat (INRIA Lille, FR)

Noah Van Es behaalde zijn MSc in Computer Science aan de Vrije Universiteit Brussel, summa cum laude. Hij begon vervolgens aan zijn doctoraat aan het Software Languages Lab onder begeleiding van Prof. dr. Coen De Roover. Zijn onderzoeksinteresses liggen voornamelijk in statische programma-analyse via abstracte interpretatie. Zijn werk resulteerde in 7 peer-reviewed publicaties als eerste auteur, die hij presenteerde op verscheidene internationale conferenties. Daarnaast werkte hij mee aan 4 andere peer-reviewed publicaties. Tijdens zijn PhD werkte hij ook aan de ECOPIPE en BUGATTI onderzoeksprojecten, en begeleidde hij 7 bachelor- en 5 mastertheses.

Abstract van het doctoraatsonderzoek

Abstracte garbage collection (GC) is de toepassing van garbage collection op abstracte interpreters. Het is geweten dat abstracte GC zowel de precisie als de performantie van de interpreter significant verbetert. In dit werk focussen we op de efficiënte integratie van abstracte GC.

Om de precisie- en performantiewinsten te maximaliseren moet abstracte GC vaak toegepast worden; tegelijkertijd brengen regelmatige toepassingen hun eigen performantiekost met zich mee. Wij stellen een nieuwe aanpak voor abstracte GC voor die gebruik maakt van het tellen van referenties (abstract reference counting). Deze aanpak kan meteen alle garbage elimineren, waardoor de precisie- en performantiewinsten van abstracte GC gemaximaliseerd worden. Tegelijkertijd hoeft het de heap niet frequent te doorlopen, waardoor het geen significante kost toevoegt. Om ervoor te zorgen dat alle garbage verwijderd is, voegen we het opsporen van cycli toe aan abstract reference counting door referenties te tellen op niveau van de sterk samenhangende componenten. De empirische evaluatie toont dat de resulterende aanpak zowel de optimale precisie als een significant betere performantie heeft vergeleken met de bestaande aanpakken voor abstracte GC.

Ten tweede bekijken we de integratie van abstracte GC voor abstracte interpreters die niet expliciet de stack modelleren (zoals abstract definitional interpreters). Bestaande technieken benutten niet de volledige voordelen van zowel abstracte GC als pushdown control flow, en/of vereisen verregaande aanpassingen aan de abstracte interpreter. Daarnaast ondervinden ze allemaal problemen met schaalbaarheid, aangezien abstracte GC niet compatibel is met global store widening. Wij presenteren delta store semantiek (DSS), een nieuwe maar eenvoudige aanpak om abstracte GC te integreren in abstract definitional interpreters. DSS geeft een delta store (die de veranderingen aan de originele store representeert) terug in plaats van de gewijzigde store als deel van het evaluatieresultaat. DSS behoudt niet alleen de voordelen van pushdown control flow en abstracte GC, maar behaalt zelfs nog grotere precisiewinsten (vanwege de agressievere GC). We bevestigen dit zowel formeel als empirisch. Om schaalbaarheidsproblemen te voorkomen, stellen we een nieuwe vorm van store widening voor DSS voor die zowel flow-sensitiviteit als een (gelimiteerde) vorm van abstracte GC toelaat.

Tenslotte tonen we hoe een bestaand effect-gedreven werklust algoritme gegeneraliseerd kan worden om efficiënt het fixpunt voor DSS-gebaseerde analyses (zowel met als zonder abstracte GC) te berekenen. Om de performantie nog meer te verbeteren ontwerpen en implementeren we ook een parallelle versie van dit algoritme. Onze empirische evaluatie toont dat dit algoritme goed schaalbaar is op multi-core architecturen voor DSS-gebaseerde analyses.