

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

Jens Van der Plas

ter behaling van de graad van Doctor in de wetenschappen

Titel van het proefschrift:

**Incremental Static Program Analysis through
Reified Computational Dependencies**

Promotoren:

Prof. dr. Coen De Roover (VUB)

**Prof. dr. Quentin Stiévenart (Université
du Québec à Montréal, CA)**

De verdediging heeft plaats op
**maandag 4 november 2024 om 17u in
promotiezaal D.2.01**

De verdediging kan ook online gevolgd
worden. Gelieve een e-mail te sturen naar
jevdplas@vub.be voor meer informatie.

Samenstelling van de jury

Prof. dr. Dominique Maes (VUB, voorzitter)

Prof. dr. Jens Nicolay (VUB, secretaris)

Prof. dr. Wolfgang De Meuter (VUB)

Prof. dr. Lynn Houthuys (VUB)

Prof. dr. Görel Hedin (Lund University, SE)

Prof. dr. Ben Hermann (Technische
Universität Dortmund, DE)

Curriculum vitae

Jens Van der Plas begon zijn studies aan de VUB in 2014, waar hij in 2019 zijn masterdiploma behaalde. Nadien startte hij een doctoraat aan het Software Languages Lab onder toezicht van Prof. dr. Coen De Roover en werd hem een FWO-beurs toegekend. Het onderzoek van Jens bevindt zich in het domein van de incrementele statische programma-analyse. Hij is coauteur van acht peer-reviewed publicaties bij internationale conferenties en in een internationaal journal en was drie keer eerste auteur. Jens begeleidde vijf bachelorproeven en evenveel masterproeven. Hij vertegenwoordigt zijn collega's in de faculteitsraad en in het faculteitsbestuur van zijn faculteit.

Abstract van het doctoraatsonderzoek

Tijdens de laatste decennia werden computers een onmisbaar deel van de moderne samenleving. Gezien de programma's die op deze computers uitgevoerd worden een essentiële rol spelen in het alledaagse leven, bijvoorbeeld in bankieren en communicatie, is het cruciaal dat ze betrouwbaar zijn. Om deze reden begonnen ontwikkelaars op statische analyses te vertrouwen om een breed scala aan programma-eigenschappen te verifiëren zonder het programma daadwerkelijk uit te voeren. Statische analyses worden doorgaans in moderne "integrated development environments" en "continuous integration"-systemen geïntegreerd om de softwarekwaliteit te bewaken gedurende het hele ontwikkelingsproces.

Gedurende het ontwikkelingsproces maken softwareontwikkelaars voortdurend kleine veranderingen aan het programma. Het is daarom niet enkel belangrijk dat een statische analyse precieze feedback geeft, maar ook dat deze efficiënt haar feedback updatet telkens wanneer het programma aangepast wordt. Om deze reden vermijdt een incrementele statische analyse het volledig herberekenen van het analyseresultaat. In plaats daarvan hergebruikt en updatet ze het voorgaande resultaat telkenmale het programma gewijzigd wordt, waardoor kostbare analysetijd bespaard wordt.

Het is uitdagend om een incrementele analyse te ontwikkelen die voldoende krachtig is om een breed scala aan programma-eigenschappen te verifiëren en die de complexe eigenschappen van hedendaagse programmeertalen kan ondersteunen. In dit werk presenteren we een nieuwe generieke methode om dergelijke analyses te construeren door gereïficeerde computationele afhankelijkheden te gebruiken. We tonen hoe een analyse die de computationele afhankelijkheden in het geanalyseerde programma reïficeert incrementeel gemaakt kan worden. Door op deze afhankelijkheden te steunen, kan onze incrementele analyse de impact van een verandering begrenzen tot de getroffen delen van het analyseresultaat. Daarenboven stellen we minimale vereisten aan de analyse zelf, hetgeen onze incrementalisatiemethode breed toepasbaar maakt.

Om de precisie van de resulterende incrementele analyse te bewaren, introduceren we drie complementaire strategieën voor resultaatrevocatie die het precisieverlies beperken. Deze strategieën zijn ontworpen rond het kernidee van het afwisselen van revocatie en