



Universiteit
Leiden
The Netherlands

Static analysis of unbounded structures in object-oriented programs

Grabe, I.

Citation

Grabe, I. (2012, December 19). *Static analysis of unbounded structures in object-oriented programs*. Uitgeverij BOXPress, Oisterwijk. Retrieved from <https://hdl.handle.net/1887/20325>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/20325>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/20325> holds various files of this Leiden University dissertation.

Author: Grabe, Immo

Title: Static analysis of unbounded structures in object-oriented programs

Issue Date: 2012-12-19

Samenvatting

Formele methoden vormen de wiskundige basis voor het redeneren over de correctheid van computer programma's. Optimaal gebruik van recente ontwikkelingen in hardware, met name de ontwikkeling van multicore processoren, vereist software die gestructureerd is op basis van parallelle processen. Dergelijke structurering echter vergroot de complexiteit van software die alleen door gebruik van formele methoden te beheersen is.

In dit proefschrift worden verschillende formele methoden voor de analyse van verschillende vormen van parallelle object georiënteerde software geïntroduceerd en bestudeerd. Centraal hierbij staan programmeerconcepten voor de beschrijving van hoe nieuwe processen dynamisch worden gecreëerd en hoe processen communiceren en synchroniseren.

In het eerste gedeelte van dit proefschrift wordt eerst een methode beschreven voor de analyse van hoe objecten gecreëerd worden in talen als Java. Object creatie vormt één van de basismechanismen van object georiënteerde software voor het programmeren van dynamische structuren en ligt ten grondslag aan de creatie van nieuwe parallelle processen in de taal Java .

Vervolgens wordt in dit proefschrift een methode beschreven voor het redeneren over de synchronisatie van parallelle processen in Java. Een belangrijke eigenschap van de correcte synchronisatie van parallelle processen is de afwezigheid van "deadlock", een situatie waarin een aantal processen op elkaar wachten en niet verder uitgevoerd kunnen worden. Voor een volledige automatisering van de beschreven methode voor het detecteren van dergelijke "deadlocks" wordt er geabstraheerd van data en beperkt tot een vast eindig aantal parallelle processen.

De complexiteit van parallelle processen zoals beschreven in de taal Java komt grotendeels doordat deze processen losgekoppeld zijn van de objecten zelf. De data van objecten wordt door de parallelle processen gedeeld waardoor het in het algemeen zeer moeilijk is het effect van deze processen te analyseren. Nieuwe object georiënteerde programmeertalen als Erlang en Scala beschrijven

daarentegen processen als het gedrag van een object zelf. Het resulterende model van een object staat bekend als een "actor" of "active object". In het derde en laatste gedeelte van het proefschrift wordt een calculus geïntroduceerd voor de algemene beschrijving van het gedrag van systemen bestaande uit verachillende "actors". Voorts wordt een methode beschreven voor de analyse van verschillende vormen van communicatie en synchronisatie tussen "actors".