Universiteit Leiden
The Netherlands

Asynchronous Programming in the Abstract Behavioural Specification Language

Azadbakht, K.

Citation
Azadbakht, K. (2019, December 11). *Asynchronous Programming in the Abstract Behavioural Specification Language*. Retrieved from https://hdl.handle.net/1887/81818

Version:	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/1887/81818

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

Cover Page

Universiteit Leiden

Leiden University Repository

The handle http://hdl.handle.net/1887/81818 holds various files of this Leiden University dissertation.

Author: Azadbakht, K.
Title: Asynchronous Programming in the Abstract Behavioural Specification Language
Issue Date: 2019-12-11

Samenvatting

Veelvoorkomende objectgeoriënteerde programmeertalen maken gebruik van "multithreading" als het gaat om het modelleren van parallelle berekeningen: er zijn meerdere leidraden aan de hand waarvan meerdere aan elkaar parallellopende berekeningen worden uitgevoerd. Echter is het redeneren over de correctheid van programma's met meerdere leidraden erg lastig. Bovendien is het evenredig verdelen van rekenwerk over rekenkernen complex: het leidraadmodel benut mogelijk niet alle beschikbare rekencapaciteit van parallelle processoren en is lastig horizontaal op te schalen. Computerchipfabrikanten maken tegenwoordig chips met meer rekenkernen dan voorheen. Zo zijn chips, met duizenden onafhankelijke rekenkernen geëtst op dezelfde siliciumplaat, niet meer ondenkbaar. Huidige programmeertechnieken gebaseerd op het leidraadmodel, kunnen niet op eenvoudige wijze gebruikmaken van de toename in rekencapaciteit van dergelijke chips, tenzij programma's hiervoor specifiek zijn ontworpen: dit vergt veel bijzondere operaties, op laag niveau in de architectuur, om toename in parallellisme te bevangen. Het alternatief is een andere programmeertechniek, waarbij de ideeën van een leidraad van een berekening en van een zekere identiteit van een object zijn verenigd. Deze programmeertechniek wordt ook wel programmeren met "actieve objecten" genoemd.

De ABS-taal (ABS staat voor "Abstract Behavioural Specification") is bedoeld voor het ontwerpen van *uitvoerbare* wiskundige modellen van parallelle en gedistribueerde, objectgeoriënteerde computersystemen. In ABS maakt men gebruik van actieve objecten. De taal is gedefinieerd in termen van een formele operationele semantiek, die een verscheidenheid aan statische- en dynamische analysetechnieken mogelijk maakt: o.a. het detecteren van wederzijdse uitsluiting ("deadlock").

Het uiteindelijke doel van dit proefschrift is het uitbreiden van het programmeermodel en de bijbehorende analysetechnieken in ABS. Dit proefschrift is opgedeeld in vier delen op basis van verschillende resultaten: deel I beschrijft vliegensvlug de ABS-taal, benodigd voor de andere delen. In deel II passen we een uitbreiding, viz. een beperkte vorm van gemeenschappelijke geheugencellen, toe om een parallel en gedistribueerd model van het "preferential attachment"-algoritme te geven, dat men gebruikt voor simulatie van grootschalige sociale netwerken met bepaalde wiskundige eigenschappen. In deel III formaliseren we uitbreidingen van ABS voor programmeren met gegevensstromen tussen processen, en meerdere leidraden binnen objecten. Tenslotte wordt in deel IV een nieuwe techniek geïntroduceerd gebaseerd

op predicatenabstractie, om ABS modellen te analyseren op vrijheid van wederzijdse uitsluiting.