Samenvatting

Wat nu? Het analyseren en optimaliseren van systeemgedrag over tijd

Het proefschrift bestaat uit drie gedeelten:

Deel I We geven een uitleg van het discrete deel van de hybride modelleertaal χ_{t}. Daarna presenteren we een aanpak om tijd te modelleren in μCRL. Een dergelijke aanpak maakt het mogelijk om getimede systemen te modelleren met een ongetimede modelleertaal, waardoor de bestaande model checking tools kunnen worden hergebruikt. Vervolgens breiden we de aanpak uit, om daaropvolgend een algemeen schema te leveren om χ_{t} specificaties te vertalen naar μCRL specificaties. Dit vertalingsschema kan worden beschouwd als een brug tussen de gebieden van performance analyse en systeemverificatie, aangezien χ_{t} hoofdzakelijk gebruikt wordt voor het eerstgenoemde gebied en μCRL over het algemeen voor het laatstgenoemde. Daarna wordt de toepasbaarheid van het vertalingsschema gedemonstreerd door een aantal χ_{t} specificaties te vertalen en te verifiëren, waaronder met name een draaitafelsysteem. Tenslotte leiden de inzichten verkregen door het ontwerpen van het vertalingsschema tot een uitbreiding van μCRL met een notie van discrete, relatieve tijd, genaamd $\mu\mathrm{CRL}^{tick}$. Er wordt aangetoond dat $\mu\mathrm{CRL}^{tick}$ specificaties kunnen worden vertaald naar μCRL specificaties, waardoor het (opnieuw) mogelijk is om de μCRL toolset te hergebruiken voor de verificatie van getimede systemen. De $\mu\mathrm{CRL}^{tick}$ aanpak bouwt voort op eerdere voorstellen om tijd te modelleren met een ongetimede procesalgebra door gebruiksgemak voor de modelleerder te benadrukken, wat wil zeggen dat de modelleerder zich niet hoeft te bekommeren om de correctheid van het tijdmechanisme in een specificatie, en door tijdstappen van variabele grootte te verwezenlijken.

Deel II Dit deel begint met het bieden van een uniforme presentatie van de meest prominente algoritmen voor het doorzoeken van toestandsruimten in het gebied van Directed Model Checking. Veel van deze algoritmen komen voort uit het gebied van de Kunstmatige Intelligentie, en maken vaak gebruik van extra informatie over het probleem in kwestie, zodat de zoektocht kan worden geleid naar interessante gedeelten van de toestandsruimte. In de presentatie worden de overeenkomsten tussen de

zoekmethoden benadrukt, waardoor een kader wordt geschetst waarbinnen veel zoekmethoden kunnen worden geplaatst. Het overzicht wordt afgerond met het overwegen van het sturen van een zoektocht op basis van acties, en het voorstellen van een verdere generalisatie van best-first search, waarbij de zoekmethode kan bestaan uit een aantal sequentiële fasen, wat een principe van compositionaliteit van zoekmethoden voor toestandsruimten introduceert. Tenslotte wordt er een vakwoordenlijst voorgesteld voor de termen in Directed Model Checking.

Vervolgens wordt de focus verlegd naar het modelleren en oplossen van schedulingproblemen met behulp van bestaande model checkers en hun invoertalen. De technieken die beschikbaar zijn in de model checkers SPIN (met PROMELA) en UPPAAL CORA (met geprijsde getimede automaten) voor scheduling worden besproken, en op sommige punten uitgebreid om de efficiëntie van de technieken of de kwaliteit van de resultaten te verbeteren. Daarnaast worden er nieuwe technieken voorgesteld, die geïmplementeerd zijn in de μCRL toolset.

Het beam search algoritme is het onderwerp van het volgende hoofdstuk. Traditioneel wordt beam search toegepast op zeer gestructureerde zoekbomen. Door het basisalgoritme op meerdere manieren uit te breiden kan het efficient worden toegepast op willekeurige toestandsruimten. Zowel toestandsgerelateerde (detailed) beam searches als actiegerelateerde (priority) beam searches worden ontwikkeld en besproken. Dit leidt tot een spectrum van beam searches. Door een voorgesteld mechanisme om zoekalgoritmen te vergelijken toe te passen, kunnen we vaststellen dat meerdere prominente zoekmethoden kunnen worden beschouwd als specifieke instanties van beam search in dit spectrum. Tenslotte wordt er aangetoond hoe deze beam search varianten, en een andere, uitputtende, zoekmethode voor scheduling, kunnen worden aangepast om te werken in een gedistribueerde omgeving, waarin een aantal computers samenwerken in een cluster om een toestandsruimte te genereren, of te doorzoeken.

Het deel wordt afgesloten door een aantal casestudies te presenteren waarbij de meeste voorgestelde zoekalgoritmen in de praktijk zijn toegepast. De meest opmerkelijke casestudie is een analysator van klinische chemicaliën. In alle gevallen is de efficiëntie en effectiviteit van de technieken in de μCRL toolset geanalyseerd, terwijl in één geval de technieken in de μCRL toolset zijn vergeleken met technieken beschikbaar in de model checker SPIN.

Deel III Getimed gedrag van systemen kan worden vergeleken door middel van getimede versies van bisimilariteitsrelaties. We kijken naar de eigenschappen van getimede vertakkende bisimilariteit, en wijzen erop dat de bestaande definitie van Van der Zwaag niet transitief is in een absolute, continue tijdomgeving. Dientengevolge moet de definitie worden uitgebreid om te verzekeren dat het een equivalentie is in een absolute, continue tijdomgeving. Een sterkere notie wordt voorgesteld (sterker in de zin dat het minder processen met elkaar relateert), en er wordt bewezen dat deze getimede vertakkende bisimilariteit inderdaad een equivalentie is, zelfs als het tijdsdomein

continu is. Voorts laten we zien dat in het geval van een discreet tijdsdomein de notie van Van der Zwaag en onze sterkere notie overeenkomen. Zoals Appendix C laat zien is het gepresenteerde tegenvoorbeeld voor transitiviteit ook van toepassing op de notie van getimede vertakkende bisimilariteit van Baeten & Middelburg in het geval van een continu tijdsdomein. Dus die notie vestigt ook niet een equivalentierelatie.

Daarna wordt er een gewortelde versie van de uitgebreide getimede vertakkende bisimilariteit gedefinieerd, en wordt er bewezen dat het een congruentie is over een procesalgebra met parallellisme, succesvolle terminatie, en deadlock. Op een aantal punten wijkt het bewijs af van het gebruikelijke congruentiebewijs voor ongetimede vertakkende bisimilariteit. Vanwege de aanwezigheid van succesvolle terminatie is er bijvoorbeeld een groot aantal gevallen. Feitelijk is het congruentiebewijs voor de parallelle-compositieoperator beperkt tot een situatie zonder succesvolle terminatie, aangezien het aantal gevallen in een bewijs met succesvolle terminatie gewoonweg te groot is. Bovendien wordt er gedemonstreerd dat de standaardaanpak voor ongetimede vertakkende bisimilariteit, namelijk het nemen van de kleinste congruentiesluiting en bewijzen dat deze een vertakkende bisimulatie levert, tekort schiet in een getimede omgeving.