# Verslag Tinlab Advanced Algorithms

**Galvin Bartes 0799967**
**176-671**

**19 november 2023**

# Inhoudsopgave

# 1 Inleiding

In dit verslag ga ik in op de kennis en achtergrondinformatie die nodig is voor het toepassen van Time-based bodel technieken. Door de toenemende complexiteit van systemen is het gebruik van modellen en de toepassing van timebased model checking op industriele controle systemen een manier van modelleren van het systeem en de requirements zodat er een bijdagre kan worden geleverd aan de acceptatie van simulatie-/modeltechniek voor de industrie.[**?**]. De behandelde onderwerpen zijin requirements-engineering, computational Tree Logic, propositielogica en predicatenlogica.

# 2 Theoretisch kader

In dit hoofdstuk houdt ik me bezig met de bestudering van rampen aan de hand van het vier-variabelen model maakt het analyseren mogelijk van rampsituaties. Van een aantal rampen is een beschrijving gegeven met datum, plaats en oorzaak. De analyse van de 4-variabelen modellen zal gebruikt worden voor de requirementsdefinitie, ontwerp en ontwikkeling van het sluismodel.

## 2.1 Begrippen, tools en literatuur

**Wat is uppaal** Uppaal is een geïntegreerde toolomgeving voor het modelleren, simuleren en verifiëren van real-time systemen, gezamenlijk ontwikkeld door Basic Research in Computer Science aan de Universiteit van Aalborg in Denemarken en de afdeling Informatietechnologie aan de Universiteit van Uppsala in Zweden. Het is geschikt voor systemen die kunnen worden gemodelleerd als een verzameling niet-deterministische processen met een eindige controlestructuur en klokken met reële waarde, die communiceren via kanalen of gedeelde variabelen . Typische toepassingsgebieden zijn met name real-time controllers en communicatieprotocollen, waarbij timingaspecten van cruciaal belang zijn.

**Wat is statistical model checking?** Dit verwijst naar verschillende technieken dfie worden gebruikt voor de monitoring van een systeem. Daarbij wordt vooral gelet op een specifieke eigenschap. Met de resultaten van de statsitieken wordt de juistheid van een ontwerp beoordeeld. Statistisch model checking wordt onder andere toegepast in systeembiologie, software engineering en industriele toepassingen. [?] [?]

**Waarom gebruiken we statistisch model checking?** Om de bovenstaande problemen te overwinnen stellen we voor om te werken met Statistical Model Checking , een aanpak die onlangs is voorgesteld als alternatief om een uitputtende verkenning van de toestandsruimte van het model te vermijden. Het kernidee van de aanpak is om een aantal simulaties van het systeem uit te voeren, deze te monitoren en vervolgens de resultaten uit het statistische gebied te gebruiken (inclusief het testen van sequentiële hypothesen of Monte Carlo-simulaties) om te beslissen of het systeem aan de eigenschap voldoet of niet. mate van vertrouwen. Van nature is SMC een compromis tussen testen en klassieke modelcontroletechnieken. Het is bekend dat op simulatie gebaseerde methoden veel minder geheugen- en tijdintensief zijn dan uitputtende methoden, en vaak de enige optie zijn. [?]Alternatieve tools voor Uppaal zijn Asynchronous Events,Vesta en MRMC.

**MODE CONFUSION** Mode confusion treed op als geobserveerd gedrag van een technisch systeem niet past in het gedragspatroon dat de gebruiker in zijn beeldvorming heeft en ook niet met voorstellingsvermogen kan bevatten.

**Wat is automatiseringsparadox** Gemak dient de mens. Als er veel energie wordt gestoken in de ontwikkeling van hulmiddelen die taken van werknemers overemen heeft dat tot resultaat dat veel productieprocessen worden geautomatiseerd. De vraag is dan of vanuit mechnisch wereldpunt de robot niet de rol van de mens overneemt en of de mens nog de kwaliteiten heeft om het werk zelf te doen. [?] [?] [?]

**4 variabelen model** Er zijn veel veiligheidskritische computersystemen nodig voor het bewaken en besturen van fysieke processen processen. Het viervariabelenmodel, dat al bijna met succes in de industrie wordt gebruikt veertig jaar, helpt het gedrag van en de grenzen tussen het fysieke te verduidelijken processen, invoer-/uitvoerapparaten en software. [?]

Het 4 variabelen model kort toegelicht Monitored variabelen: door sensoren gekwantificeerde fenomenen uit de omgeving, bijv temperatuur

Controlled variabelen: door actuatoren bestuurde fenomenen uit de omgeving Gecontroleerde variabelen kunnen bijvoorbeeld de druk en de temperatuur zijn in een kernreactor, terwijl gecontroleerde variabelen ook visuele en hoorbare alarmen kunnen zijn als het uitschakelsignaal dat een reactorsluiting initieert; wanneer de temperatuur of druk bereikt abnormale waarden, de alarmen gaan af en de uitschakelprocedure wordt gestart

Input variabelen: data die de software als input gebruikt Hier modelleert IN de ingangshardware-interface (sensoren en analoog-naar-digitaal-omzetters) en relateert waarden van bewaakte variabelen aan waarden van invoervariabelen in de software. De invoervariabelen modelleren de informatie over de omgeving die beschikbaar is voor de software. Bijvoorbeeld, IN zou een druksensor kunnen modelleren die temperatuurwaarden omzet in analoge spanningen; Deze spanningen worden vervolgens via een A/D-omzetter omgezet in gehele waarden die zijn opgeslagen in een register dat toegankelijk is voor de computer software.

Output variabelen: data die de software levert als output De uitgangshardware-interface (digitaal-naar-analoog converters en actuatoren) is gemodelleerd door OUT, dat waarden van de uitvoervariabelen van de software relateert aan waarden van gecontroleerde variabelen. Een uitvoervariabele kan bijvoorbeeld een Booleaanse variabele zijn die door de software is ingesteld met de begrijpen dat de waarde true aangeeft dat een reactorsluiting zou moeten plaatsvinden en de waarde false geeft het tegenovergestelde aan

**6 Variable model** Een uitbreiding van een 4-variabelen model is het 6-variabelen model. Optitatieve statements omschrijven de omgeving zoals we het willen zien vanwege de machine. Indicatieve statements omschrijven de omgeving zoals deze is los van de machine. Een requirement is een optitatief statement omdat ten doel heeft om de klantwens uit te drukken in een softwareontwikkel project. Domein kennis bestaut uit indicatieve uitspraken die vanuit het oogpunt van software ontwikkeling relevant zijn. Een specificatie is een optitatief statement met als doel direct implementeerbaar te zijn en ter ververondersteuning van het nastreven en realiseren van de requirements. Drie verschillende type domeinkennis: domein eigenschappen, domein hypothesen, en verwachtingen. Domein eingenschappen zijn beschrijvende statementsover een omgeving en zijn feiten. Domein hypotheses zijn ook beschrijvende uitspraken over een omgeving, maar zijn aannames. Verwachtingen zijn ook aannames, maar dat zijn voorschrijvende uitspraken die behaald worden door actoren als personen, sensoren en actuators.

## 2.2 Rampen

Voor deze studie is onderzoek gedaan naar verschillende rampen aan de hand van het vier variabelen model. Elke ramp op deze manier categoriseren kan ons helpen te bepalen in hoeverre requirements een rol kunnen spelen in de veiligheid van ons model.

### 2.2.1 Therac-25

**Beschrijving** In de periode van Juni 1985 and Januarie 1987 zijn er meerdere ongelukken met dodelijke afloop door de implementatie van de Therac-25 bij de behandelig van huidkanker. Dit apparaat gebruikt elektronen om stralen met hoge energie te creëren die tumoren kunnen vernietigen met minimale impact op het omliggende gezonde weefsel.

**Datum en Plaats** In de periode van Juni 1985 and Januarie 1987

**Oorzaak** Onderzoekers constateren dat er fouten zijn gemaakt tijdens de (her-)implementatie van systemen uit eerdere productiemodellen. Terwijl de therac 20 afhankelijk was van mechanische vergrendelingen werd er bij de therac-25 software gebruikt. Onderzoekers komen daarom tot de volgende conclusies Software problemen zijn onder andere:

- slechte software engineering/designing praktijken
- er is een machine gebouwd dat afhankelijk is van software voor veilgheidsoperaties
- de fout in de code is niet zo belangrijk als een geheel onveilig ontwer
- het reinigen van de buigmagneetvariabele in plaats van aan het uiteinde van het frame
- raceconditionering om aan te geven dat het invoeren van het recept nog steeds aan de gang is
- reactie van de gebruiker
- slechte subroutines voor schermvernieuwing die rommel en foutieve informatie op de werkende console achterlieten
- Problemen met het laden van tapes bij het opstarten, waarbij het gebruik van photom-tabellen werd uitgesloten om het interlock-systeem te activeren in het geval van een laadfout in plaats van een checksum

Onderzoekers zijn van mening dat de terkortkomingen in medische apparatuur niet geheel en altijd te verwijten zijn aan softwareproblemen. Zo is er ook een rol weggelegd voor fabrikanten en overheden. Klankbordgroepen klagen over het tekort aan software-evaluaties en een tekort aan hard-copy audit trials om foutmeldinen in beeld te krijgen. [**?**]

### 2.2.2 Ethiopian Airlines Flight 302,boeing 737 crashes

**Beschrijving** Ethiopische Airlines-vlucht 302 was een geplande internationale passagiersvlucht van Bole International Airport in Addis Abeba, Ethiopië naar Jomo Kenyatta International Airport in Nairobi, Kenia. Op 10 maart 2019 stortte het Boeing 737 MAX 8-vliegtuig dat de vlucht uitvoerde zes minuten na het opstijgen neer nabij de stad Bishoftu, waarbij alle 157 mensen aan boord omkwamen.

Vlucht 302 is het dodelijkste ongeval van Ethiopië Airlines tot nu toe en overtreft de fatale kaping van vlucht 961, resulterend in een crash nabij de Comoren in 1996. Het is ook het dodelijkste vliegtuigongeluk dat zich in Ethiopië heeft voorgedaan, en overtreft de crash van een Antonov An-26 van de Ethiopische luchtmacht in 1982, waarbij 73 mensen omkwamen.

Dit was het tweede MAX 8-ongeluk in minder dan vijf maanden na de crash van Lion Air-vlucht 610. Beide crashes waren aanleiding voor een twee jaar durende wereldwijde langdurige aan de grond

houden van het vliegtuig en een onderzoek naar de manier waarop het vliegtuig werd goedgekeurd voor passagiersvervoer.

**Datum en Plaats**

**Oorzaak** De technische oorzaak ligt bij het MCAS flight control system. Dit systeem werd geimplementeerd om kosten te reduceren en opleidingen voor piloten in te korten. Deze single point of failure [?] werd getriggerd door een enkele angle-of-attack sensor[?]. Bij de nieuwe boeing 737 max model werden tests uitgevoerd in volledige flight simulators. Nieuwe regels van het FAA instituut vereisten ondersteuning bij het uitvoeren van enkele manouvres. Tijdens testvluchten uitgevoerd binnen een jaar voor certificatie werd het pitch-up fenomeen geconstateerd waarop het mcas systeem werd aangepast. In het systeem zaten nu de volgende fouten.

- MCAS wordt getriggerd woor enkele sensor zonder vertraging
- Het ontwerp staat toe dat in situaties waar de angle-of-attack fout is de mCAS wordt geactiveerd
- systeem kreeg onnidig bevoegdheid controle om de neus bij te sturen
- Waarschuwingslicht bij fouten in de angle-of-attack werkte niet door softwarefout. Het werd ook niet kritisch ebvonden door ethiopian airlines. geplande updates door boeing pas in 2020
- een losstande fout in de microprocessor va de controle computer kan vergelijkbare situaties doen voorkomen zonder dat mcas wordt geactiveerd

Fouten vielen niet op omdat FAA test uitbesteedde aan Boeing. Contact tussen de organisaties FAA en Boeing verliep op management niveau. Boeing instrueerde niet alle piloten ovver MCAS. En het MCAS systeem wwerd gezien als een achtergrond systeem.

Uit onderzoek is op te maken dat problemen bij boeing toestellen te verwijten zijn aan:

- marktdruk
- slapshot engineering
- missiekritische toepassing van technologie
- geen regelgevingsregime voor foutieve risicoanalyse

### 2.2.3 China explosie 2015 Tianjin

**Beschrijving** De explosie zorgde voor de vernietiging van 12000 voertuigen, schade aan 17000 huize binnen een traal van 1 km. Er waren 173 doden inclusief brandweermensen. Een van de explosies zorgde voor een beving van 2.3 op de schaal van rigter. Opgeslagen materialen waren: calcium carbine, sodium nitraat, potassium nitraat, amminiak nitraat en cyanide. Ook is er veel kritiek geweest op de acties van de autoriteiten. Zo was er censuur vanuit de overheid op de journalistiek. Ook was er naar alle warschijnlijkheid sprake van corruptie. Zo bleek achteraf dat een van de grootste aandeelhouders Dong Shexuang de zoon te zijn van een oud-politiechef in Tanjin haven, genaamd Dong Pijun

**Datum en Plaats** Op 12 augustus 2015 waren er twee explosies bij de Rulthai logistiek. Deze faciliteit zorgde voor de opslag van gevaarlijke stoffen. De explosie tanjin china 12/08/2015.

**Oorzaak** De volgende factoren zouden een rol hebben gepeeld:

- Een onjuiste afbakening van het opslagmaeriaal
- Er was weinig kennis bij de autoriteiten over opslagmaterialen. Zo bleek er 7000 ton aan materiaal opgeslagen, dat is ruim 70 keer te maximaal toegestande hoeveelheid.
- Onverenigbaar grondgebruik in de nabije omgeving. Veel woonwijken met nar schatting 6000000 bewoners en 500 lokale bedrijvenin de buurt van de opslag gevaarlijke stoffen.

### 2.2.4 de malimissie

**Beschrijving** Aanwezige militair brengt slachtoffer naar de fransen, vervolgens naar de Tongolezen. Maar de kwaliteit van personeel liet te wensen over. Er werd een Nederlandse arts overgevlogen. De slachtoffers werden overgevlogen naar Gao omvervolgens te worden oergevolgen naar Nederland. Conclusie van de OVV[?]

- Koopcontract werd niet goed doorgelezen
- Geen controle op kwaliteit en veiligheid
- Zwakke plekken in het ontwerp
- Opslag en gebruik in ongunstige condities
- De aanwezige medische voorzieningen waren nite volgends de nederlandse militaire richtlijnen
- Het ontbreek aan medische toetsing vanuit de defensie organisatie
- Twijfels die werden geuit binnen de defensieorganisae vonden geen wrrklank
- Ook het ongeval tijdens de mortieroefening was voor defensie geen aanleuiding om de medische voorzienignen te evalueren.
- De inrichting van veilige medische zorg voor nederlandse militairen in Kidal is ondergeschikt gemaakt aan de voortgang van de missie.

**Datum en Plaats** Het mortierongeluk in Mali op 06/04/2016.

**Oorzaak** Het ongeluk werd veroorzaakt door een kapot afsluitplaatje in de mortier. Tijdens de oefening werden de granaten warm in de zon. Granaat werd opgeslagen in niet gekoelde containers waardoor deze aan te hoge temeperaturen zijn blootgesteld. Dan was er vocht in de fatale granaat. Door de combinatie van vocht en warmte in de granaat werden zeer gevoelige explosieve stoffen werden gevormd. Het afsluitplaatje in de granaat bleek niet in staat om doorslag in veilige stand te voorkomen waarna de granaat explodeerde. De granaat stond niet op scherp en in afgegaan in veilige stand Uit onderzoek bleek dat de mortieren zijn aangeschaft bij de amerikanen. Gedurende de aanschafperiode zijn procedures en controles op kwaliteit en veiligheid deels nagelaten. Dit werd vermeld in het koopcontract.

### 2.2.5 schipholbrand

**Beschrijving** Bij de schipholbrand op 27/10/2005 . vielen zeker 11 doden onder mmigranten in de cellenclomplexen van schiphol-oost. Doodsoorzaak van de slachtffoffers is verstikking. Het gebrouw voldeed niet aan de eisen voor brandveiligheid, personeel was niet goed getraind voor dergelijke situaties en de hulpverlening kwam door verschillende factoren te laat op gang.

**Datum en Plaats** Bij de vleugels j en k van cellenclomplexen van schiphol-oost op 27/10/2005 .

**Oorzaak** Doordat de deur van cel 11 niet werd gesloten kreeg de brand zuurstof waardoor deze kont uitbreiden de aanwezigheid van grote hoeveelheid brandbare materialen, fungeerde als brandstof waardoor de brand zich verder kon ontwikkelen zowel binnen als buiten de cel.Doordat de dakluiken niet werken werd de rook en warmte niet afgevoerd, dit heeft de reddingsoperatie van de bewaarders belemmmerd en het mede onmogelijk gemaakt alle celdeuren te openen. Door de schilcontructie van het cellencomplex was het mogelijk dat de brand zich kon uitbreiden naar de andere cellen en naar de gang.
Dit is onopgemerkt gebleven doorat er geen speicifieke eisen werden gesteld bij de bouw van vleugels j en k. Was er te weinig aandacht voor bouwregels en onderzoek naar eeerdere branden in het cellencomplex. Bovendien waren bhv'ers niet goed opgeleid en was er geen goede afstemming met de brandweer. Er was geen evauactieplan dat voorziet in de overplaatsing van gedetineerden naar ander penetintiaire inrichtingen. En had men te weinig aandacht voor de traceerbaarheid van celbewoners en bovendien voor ademhalingsproblemen van gedetineerden.

### 2.2.6 1951

**Beschrijving** Turkish Airlines-vlucht 1951 (ook bekend als TK 1951) was een passagiersvlucht die op woensdag 25 februari 2009 neerstortte in de buurt van het Nederlandse vliegveld Schiphol. Hierbij kwamen negen van de 135 inzittenden om het leven.

Het vliegtuig, een Boeing 737-800 van de Turkse maatschappij Turkish Airlines, was op weg van het vliegveld Istanbul Atatürk naar Schiphol, maar stortte om 10:26 uur,[1] kort voor de landing op de Polderbaan, neer op een akker en brak daarbij in drie stukken.[7][8]

Het officiële onderzoek van de Onderzoeksraad voor Veiligheid wees inadequaat handelen van de piloten als hoofdoorzaak aan voor het ongeluk. Ondanks een defecte hoogtemeter en onvolledige instructies van de luchtverkeersleiding hadden de piloten het ongeluk kunnen voorkomen, aldus de Onderzoeksraad.[

**Datum en Plaats** Op woensdag 25 februar 2009 voltrok zich de ramp met een toestel van turkisch airlines tijdensnvlucht 1951.

**Oorzaak** De automatische reactie van het toestel werd getriggerd door een fout gevoelige radio altimeter waardoor de automatische gashendel de energiemotot op actief stelde. Inadequaat handelen van de piloten ondanks een defecte hoogtemeter en onvolledige instructies van de luchtverkeersleiding

### 2.2.7 slmramp

**Beschrijving** Toen het toestel op 07/06/1989 de Anthony Nesty Zanderij naderde, was het daar, anders dan het weerbericht had voorspeld, mistig. Het zicht was evenwel niet zo slecht dat er niet

op zicht kon worden geland. Gezagvoerder Will Rogers besloot echter via het Instrument Landing System (ILS) te landen, hoewel dit niet betrouwbaar was en hij voor zo'n landing ook geen toestemming had. De gezagvoerder brak drie landingspogingen af. Bij de vierde poging negeerde de bemanning de automatische waarschuwing (GPWS) dat het toestel te laag vloog. Het toestel raakte op 25 meter hoogte twee bomen. Het rolde om de lengteas en stortte om 04.27 uur plaatselijke tijd ondersteboven neer.

**Datum en Plaats** Toen het toestel op 07/06/1989 de Anthony Nesty Zanderij naderde

**Oorzaak**

Uit onderzoek bleek dat de papieren van de bemanning niet in orde ware door nalatigheid in de crew-member screening.Geconcludeerd werd dat de gezagvoerder roekeloos had gehandeld door voor een ILS-landing te kiezen terwijl hij daar geen toestemming voor had, en door onvoldoende op de vlieghoogte te hebben gelet. De SLM werd verweten de kwalificaties van de bemanning onvoldoende te hebben gecontroleerd. De Oorzaak van de ramp bleek achteraf het roekeloos besturen door de kapitein onder de minimum hoogte leidde tot collissie met een boom.

### 2.2.8 Tsjernobyl

**Beschrijving** De mislukte veiligheidscontrole die 26 april 1986 01.24 uur in de SovjetUnie leiddte tot explosies in een van de reactoren in de kerncentrale. De reactoren hadden geen veiligheidsomhulling en de reactor bevat grote hoeveelheden brandbaar grafiet. Door de explosie en de brand kwamen er radioactieve stoffen vrij.het gaat helemaal mis in de kernreactor 4. De warmteproductie nam toe met een explosie tot gevolg. 31 mensen kwamen om, waaronder veel mensen dagen later door stralingsziekte.

**Datum en Plaats** De ramp van Tjernobyl voltrok zich op 26/04/1986 [?].

**Oorzaak**

Techici bij kerncentrale 4 voerden een slcht opgezet/ ontwerpen experiment uit. De kracht regulering werd uitgeschakeld evenals veiligheidssystemen. Door een bedieningsfout in een testprocedure werd het vermogen van de koelinstallaties negatief beinvloed. Mede door een ontwerpfout in de noodstopprocedure kon in het systeem niet snel genoeg schakelen om remmende invloed uit te oefenen op het toenemende vermogen van de reactorkernen. Met brand en explosies tot gevolg. [?] [?] [?] [?] [?]

### 2.2.9 Safety critical systems

De kennis van de mens is veranwoordelijk voor jet scheppen van kennis. Opdoen van kennis van veiligeheidsintegriteit ten overstaan van ontwetendheid. Kennisdoelen zijn daarom altijd belangrijk en kunnen worden onderscheiden in 3 domeinen: cognitief, affectief en psychomotorisch.[?] Een Veiligheidsanalyse is een manier voor het evalueren van ongelukken en risicos op verschillende niveaus. Met een veiligheidsanalyse wordt gekeken naar mensen en systemen die worden blootgesteld aan een gevaar/risico en de mogelijkheid deze te minimaliseren. Veiligheid is moeilijk te definiëren omdat het een vaag concept is. Een veiligheidsanalist moet kennis hebben van het systeem, maar ook ervaring met het systeem en vooral hoe een fout in het systeem zich voordoet. Volgens Stoney[?] is veiligheid een aspect dat de veiligheid van mensenleven of de omgevig niet in gevaar brengt. De interactie tussen systeem en omgeving levert kennis op door ad-hoc ervaring. Maar ook is kennis van abstractie en systeemdoelen belangrijk. Het

kwalificeren van informatie moet ook op waarde geschat worden Een verkeerde schatting kan fataal zijn. Zelfs als een veiligheidsanalyse is gedaan waarbij risicos zijn gedefinieerd en geprioretiseerd is een definitie van een veiligheidsniveau nodig. Soms zijn hier richtlijnen voor en soms ook niet. Er zijn voorbeelden van tools en technieken te gebruiken voor een veiligheidsanalyse. En dat zijn

1. checklists voor critical safety items
2. fault tree analysis
3. event tree analysis
4. failure modes en critical effects analysis (FMECA, FMET)
5. hazard operability studies

[?] [?] [?] [?] [?] [?] [?].
Modellen

## 2.3 De Kripke structuur

## 2.4 Soorten modellen

### 2.4.1 LTL

### 2.4.2 CTL

### 2.4.3 buchi Automata

## 2.5 Tijd

### 2.5.1 Een begrensde responseigenschap

## 2.6 Guards en invarianten

Het systeem is een samenstelling van parallelle processen elk proces is gemodelleerd als een automaat de automaat heeft een aantal posities, positieveranderingen worden gedaan met behulp van randen / overgangen. De toestand van het systeem wordt gekenmerkt door de huidige positie van elke automaat, waarden van variabelen, en klokstatus. overgangen kunnen worden gecontroleerd met behulp van bewakers en synchronisaties guard is een voorwaarde voor variabelen en klokken aangeven wanneer een transitie mogelijk is.

Onveranderbaar . . . Conjunctie van voorwaarden van de vorm x ¡e of x ¡= e, waar x is een link naar een klok, e wordt berekend tot een geheel getal. Het kan verwijzen naar klokken, gehele variabelen, constanten.

## 2.7 Deadlock

• Deadlock een bereikbare staat kan helemaal geen acties uitvoeren – Deadlock hangt af van de reeks acties die een bereikbare staat niet kan uitvoeren • Om de impasse te behouden moet A niet alleen overschatten wat P kan doen, maar ook wat P weigert

## 2.8 Zeno gedrag

zeno gedrag: de mogelijkheid dat in een eindige hoeveelheid tijd een oneindig antal handelingen kan worden verricht. Bijvoorbeeld tijdens het nivelleren Bij het opstellen van schepen Bij het laten wachten van schepen Bij het invaren van schepen

# 3 Logica

## 3.1 Propositielogica

In de propositielogica onderzoekt men het waarheidsgehalte van samengestelde uitspraken aan de hand van elementaire proposities en logische voegwoorden. Wij noemen de woorden 'en', 'of', 'als . . . dan . . . ', 'impliceert' logische voegwoorden. Hoewel het taalkundig geen voegwoorden zijn, noemen wij de ontkenningen 'niet' en 'geen' toch logische voegwoorden.

In de propositielogica worden proposities gemaakt van elementaire proposities en logische voegwoorden. In de symbolische propositielogica, de propositierekening, wordt een propositieformule gemaakt van variabelen (letters), logische operatoren en haakjes. Net zoals een formule in de rekenkunde, heeft het deel van een propositieformule binnen de haakjes een hogere prioriteit dan het deel buiten de haakjes. Bovendien wordt alles wat tussen haakjes staat, beschouwd als een eenheid. Om een propositieformule correct samen te stellen, moeten wij verplicht gebruik maken van de volgende grammaticale regels:

1. Een variabele is een formule;
2. Als p een formule is, dan is :p een formule;
3. Als p en q formules zijn, dan zijn (p$^q$), $(p_q), (p!q) en (p$q) formules.

In een propositieformule mogen haakjes worden weggelaten mits er geen verwarring ontstaat

Tijdens het logisch beschrijven van logica of bij het maken van formules over formules, is het mogelijk dat wij een paradoxale uitspraak doen. Zo'n uitspraak is vergelijkbaar met de volgende zin: Deze zin is onwaar De bovenstaande zin beschrijft zichzelf. Is deze zin nu waar of onwaar? Dit is een paradox. Wij moeten de uitspraken over de logica scheiden van de logica zelf. Daarom introduceren wij metasymbolen. Deze metasymbolen zijn geen onderdeel van de beschreven logica, maar een toevoeging aan de omgangstaal in dit dictaat.

Voor het bepalen van de waarheidswaarde van een formule f(p1; p2; : : : ; pn) moeten alle mogelijke combinaties van waardetoekenningen worden bepaald. Deze combinatie van waardetoekenningen vormen samen de waarheidstabel van deze formule.

## 3.2 Predicatenlogica

Hoewel de volgende redenatie in de propositielogica ongeldig is, wordt zij intu¨ıtief toch als geldig beschouwd:

1. f1 "alle republikeinen plegen geen overspel"
2. f2 "sommige overspeligen zijn president"
3. y "sommige presidenten zijn geen republikein"

De geldigheid van deze redenatie is gebaseerd op informatie, waarmee de propositielogica geen rekening mee houdt. Om deze extra informatie bij het redeneren te betrekken, moeten wij de propositielogica uitbreiden met de begrippen eigenschappen, variabelen en kwantoren. De zin "alle mensen zijn sterfelijk" geeft aan dat objecten, die de eigenschap 'menselijk zijn' hebben, blijkbaar ook de eigenschap 'sterfelijk zijn' hebben. uitspraak symbolisch:

1. sterfelijk objecten S(x)
2. menselijke objecten M(x)
3. x is een priemgetal P(x)

In de uitspraken geven wij de eigenschappen sterfelijk en priemgetal aan met de hoofdletters S en P. De variabele x stelt objecten voor. Een variabele, waarvan de waarde onbekend is, noemen wij vrije variabele. Definitie 4.1 (Predikaat) Een predikaat is een uitspraak met vrije variabelen, die een propositie wordt zodra alle vrije variabelen in dat predikaat gebonden zijn aan een waarde.

## 3.3 Kwantoren

Kwantificator wordt gebruikt om de variabele van predikaten te kwantificeren. Het bevat een formule, een soort verklaring waarvan de waarheidswaarde kan afhangen van de waarden van sommige variabelen. Wanneer we een vaste waarde aan een predikaat toekennen, wordt het een propositie. Op een andere manier kunnen we zeggen dat als we het predikaat kwantificeren, het predikaat een propositie wordt. Kwantificeren is dus een woordsoort dat verwijst naar kwantificeringen als 'alles' of 'sommige'.

Er zijn hoofdzakelijk twee soorten kwantoren: universele kwantoren en existentiële kwantoren. Daarnaast hebben we ook andere soorten kwantoren, zoals geneste kwantoren en kwantoren in standaard Engels gebruik. Kwantificator wordt voornamelijk gebruikt om aan te tonen dat voor hoeveel elementen een beschreven predikaat waar is. Het laat ook zien dat voor alle mogelijke waarden of voor sommige waarde(n) in het universum van het discours het predikaat waar is of niet.

Tot dusverre hebben wij in de voorbeelduitspraken "voor alle x" en "er is een x" gekoppeld aan alle voorkomens van x in een universum. De uitspraak "er is een x in het universum van x met de eigenschap P" wordt weergegeven als:

Soms willen wij zo'n universum beperken. Zo'n beperkt deel van een universum, een domein, is het gebied waaruit de gebonden variabele moet putten. De beschrijving van het domein D(x) wordt onder de kwantor geplaatst.

De beschrijving van het domein D(x) is zelf ook een predikaat. Het is een eigenschap D die toegekend wordt aan de variabele x. Het is mogelijk dat zo'n domein D leeg is. Men zou dat kunnen noteren met :9x D(x). Het lege domein wordt meestal aangegeven met het Q teken voor een leeg domein:

In de volgende voorbeelden wordt het domein van de kwantor beschreven met een predikaat (1 ¡ x ¡ 4). De uitspraak "alle getallen in het domein van de getallen tussen de een en de vier, zijn priemgetallen" wordt genoteerd als:

Wij kunnen dit ook schrijven als "alle getallen, indien zij tussen een en vier liggen, zijn priemgetallen". Het domein is implicatief gekoppeld aan de eigenschap P(x). Dit noemen wij een domeinoverheveling:

Zo'n overheveling van het domein kunnen wij eveneens toepassen in een uitspraak met een existenti¨ele kwantor. De uitspraak "er is een getal tussen de een en vier, dat een priemgetal is" wordt genoteerd als:

Deze uitspraak is het zelfde als "er is een getal, dat tussen de een en vier ligt en priemgetal is". Het domein wordt conjunctief gekoppeld aan de eigenschap P(x):

Als er geen domein wordt opgegeven, dan bedoelt men alle objecten in het universum van x. Zo'n universum kan eindig of oneindig veel objecten bevatten. Het studentenuniversum is bijvoorbeeld eindig, daarentegen is het universum van de re¨ele getallen oneindig. In dit hoofdstuk behandelen wij de kwantorformules over eindige universa. Gelukkig zijn wij in de informatica tevreden met eindige universa voor onze algoritmen. Eindige universa worden vaak aangegeven met symbolische grenzen, zoals bijvoorbeeld: 0;1;2: : :n. In hoofdstuk 5 worden kwantorformules afgeleid die algemeen geldig zijn, ook voor oneindige universa.

Met de universele kwantor 8x bedoelen wij alle waarden van x binnen het opgegeven domein. Als het domein niet wordt opgegeven, dan bedoelen wij een of meer waarden x uit een eindig universum. Bijvoorbeeld:

Met de existenti¨ele kwantor 9x bedoelen wij ´e´en of meer waarden van x uit het opgegeven domein. Als het domein niet wordt opgegeven, dan bedoelen wij een of meer waarden x uit een eindig universum. Bijvoorbeeld:

### 3.4 Dualiteiten

Het dualiteitsbeginsel is een soort doordringende eigenschap van de algebraïsche structuur waarbij twee concepten alleen uitwisselbaar zijn als alle resultaten in de ene formulering ook gelden in een andere. Dit concept staat bekend als dubbele formulering. We zullen unions() omwisselen in intersecties() of intersecties() in de union() en ook de universele set omwisselen voor de nulset() of nullset voor universal(U) om de dubbele verklaring te krijgen. Als we het symbool verwisselen en deze verklaring zelf krijgen, zal deze bekend staan als de zelf-duale verklaring.Het dualiteitsbeginsel is een soort doordringende eigenschap van de algebraïsche structuur waarbij twee concepten alleen uitwisselbaar zijn als alle resultaten in de ene formulering ook gelden in een andere. Dit concept staat bekend als dubbele formulering. We zullen unions() omwisselen in intersecties() of intersecties() in de union() en ook de universele set omwisselen voor de nulset() of nullset voor universal(U) om de dubbele verklaring te krijgen. Als we het symbool verwisselen en deze verklaring zelf krijgen, zal deze bekend staan als de zelf-duale verklaring.

Dualiteit, in de wiskunde, principe waarbij de ene ware uitspraak uit de andere kan worden verkregen door slechts twee woorden met elkaar te verwisselen. Het is een eigenschap die behoort tot de tak van de algebra die bekend staat als de roostertheorie en die betrokken is bij de concepten van orde en structuur die in verschillende wiskundige systemen voorkomen. Een wiskundige structuur wordt een rooster genoemd als deze op een bepaalde manier kan worden geordend (zie volgorde). Projectieve meetkunde, verzamelingenleer en symbolische logica zijn voorbeelden van systemen met onderliggende roosterstructuren, en hebben daarom ook principes van dualiteit.

## 4 Computation tree logic

### 4.1 Algemeen

Computatieboomlogica (CTL) is een vertakkingstijdlogica dat omvat zowel de propositionele connectieven als temporele verbindingen AX, EX, AU, EU, AG, EG, AF, en EF. We moeten aantonen dat een real-time programma voldoet aan de eisen opgesteld en gespecificeerd.

Een formele verificatie van de gespecificeerde requirements ofwel modeleigenschappen wordt gerealiseerd door deze te vertalen naar de query-language van de symbolic model-checker Uppaal. [?],[?].

Temporele logica is een al lang gebruikt en goed begrepen concept om softwarespecificaties en computerprogramma's te modelleren door middel van toestandsovergangsemantiek. Een pad is een

reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Een vertakkende tijdpadformule strekt zich uit over mogelijke reeksen van opeenvolgende toestanden. States hebben labels. Als de huidige toestand een label p heeft, dan is de atomaire toestandsformule p waar in die toestand. Toestandsformules worden opgebouwd uit deze atomaire formules met behulp van de Booleaanse connectieven en de padlogische kwantificatoren, met de beperking dat we een padlogische kwantificator in een toestandsformule niet kunnen ontkennen. (Negatie kan effectief worden toegepast op conjuncties en disjuncties door de negatief-normale vorm te gebruiken).

Het bewijs van de modeleigenschappen kan wiskundig worden opgesteld met behulp van kwantoren, zoals:

De kripke structuur is een set van locaties, transities, guards, klokken en data-variabelen . Temporal logic formele taal voor het specificeren en redeneren over hoe de Het gedrag van een systeem verandert in de loop van de tijd breidt de propositielogica uit met modale/temporele operatoren een belangrijk gebruik: representatie van toekomstige systeemeigenschappen gecontroleerd door een modelchecker. Informeel betekenen A en E 'langs alle paden' en 'langs ten minste één pad", respectievelijk; terwijl X, F, G en U verwijst naar 'volgende staat', 'een toekomstige staat', 'hele toekomst' staten," en "Tot." We gebruiken structurele inductie op CTL-formules om te definiëren een tevredenheidsrelatie: M,s impliceert phi dat hangt af van een transitiesysteem M = (S, $\rightarrow$, L), a toestand s van M, en een CTL-formule $\phi$. Syntaxis van CTL is opgesplitst in state- en padformules specificeer respectievelijk eigenschappen van toestanden/paden een CTL-formule is een toestandsformule State formulae: waarbij een E AP en een padformule is voor een atomaire propositie Path formulae waarbij een toestandsformule (stateformula) is Elke padkwantificator moet worden gevolgd door een temporele kwantificator in de syntactische boom van elke formule

Een overzicht van meerde temporal operators:

1. F sometime in the Future, The future time operator (F) is used to specify that some property eventually holds at some state in our path

2. G Globally in the future, The global operator (G) is used to specify that some property holds for all states of a particular path.

3. X neXtime, The next time operator (X) is used to specify that some property holds in the second state of a path.

4. U until, The until operator (U) is a binary operator, and is used to specify that the first property holds in all states preceding the one where the second property is satisfied

5. R the release operator, The release operator (R) is also a binary operator, and is used to specify that the second property holds in all states along a path up to and including the first state that satisfies the first property. The first property is not required to be eventually satisfied.

Path quantifiers:

1. E there Exists a path

2. A in All paths

## 4.2 Operator: AG

AG P: in all possible execution paths, P is always true.

## 4.3 Operator: EG

EGp - p holds globally. EG P: in at least one execution path, P is always true. Een pad is een reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Een vertakkende tijdpadformule strekt zich uit over een enkele reeks van opeenvolgende toestanden. In de huidige reeks heeft elke toestand een label p , en is de atomaire toestandsformule p waar in die toestand behorend tot deze reeks.

## 4.4 Operator: AF

For all finally AFp - p holds sometime in the future. • If any state s is labeled with 1, label it with AF 1.

Repeat: label any state with AF 1 if all successor states are labeled with AF 1, until there is no change. AF P: in all possible execution paths, sooner or later P will be true.

Een pad is een reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Deze vertakkende tijdpadformule strekt zich uit over meerdere paden met in de toekomst een geldige opvolgende toestand. Als voor alle paden geldt dat deze vanaf de huidige toestan er in de toekomst een toestand een label p heeft, dan is de atomaire toestandsformule p waar in die toestand.

## 4.5 Operator: EF

EF, there exists a path where finally ,from the current state, along which some state satisfies EF P: in at least one execution path, sooner or later P will be true.

We hebben hier een pad met een reeks toestanden. Waarbij ooit op dit pad een toestand kan toestandsformule worden geëvalueerd welke een label p heeft, ed de atomaire toestandsformule p waar is in die toestand.

## 4.6 Operator: AX

AXp - for all paths p holds next time. AX P: in all possible execution paths, in the next state P is true.

We spreken hier van alle mogelijke paden waarvoor geldt dat in de volgende toestand kan een toestandsformule worden als "waar"wordt geëvalueerd. States hebben labels. Als de volgende toestand een label p heeft, dan is de atomaire toestandsformule p waar in die toestand.

## 4.7 Operator: EX

EXp - for some path p holds next time. : label any state with EX 1 if one of its successors is labeled with 1 . EX P: in at least one execution path, in the next state P is true.

Een pad is een reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Een vertakkende tijdpadformule strekt zich uit over een mogelijke reeks van opeenvolgende toestanden. Waarbij geldt dat in de huidige reeks er een toestand is met een opvolgende state met een label p heeft, en is de atomaire toestandsformule p waar in die toestand.

## 4.8 Operator: p U q

p holds until q holds. A[P U Q]: in all possible execution paths, P is true * until Q is true (at least once). E[P U Q]: in at least one execution path, P is true * until Q is true (at least once).

Een pad is een reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Een vertakkende tijdpadformule strekt zich in geval van A[P U Q]: uit over alle reeksen van opeenvolgende toestanden. Waarvoor geldt dat als de toestand een label p heeft, dan is de atomaire toestandsformule p waar in die toestand tot aan de toestand waarin een toestand het label q heeft.

Een pad is een reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Een vertakkende tijdpadformule strekt zich uit over een enkele reeks van opeenvolgende toestanden. Waarvoor geldt dat als de toestand een label p heeft, dan is de atomaire toestandsformule p waar in die toestand tot aan de toestand waarin een toestand het label q heeft.

## 4.9 Operator: p R q

R ("release") is the logical dual of the U operator. The operator requires that the second argument must hold up to and including the first state where the first argument holds. The first argument is not required to become true eventually.

• A (All): has to hold on all paths starting from the current state. • E (Exists): there exists at least one path starting from the current state where holds.

Een pad is een reeks toestanden. Bij elke toestand kan een toestandsformule worden geëvalueerd. Een vertakkende tijdpadformule strekt zich uit over een mogelijke reeks van opeenvolgende toestanden. States hebben labels. Als de huidige toestand een label q heeft, dan is de atomaire toestandsformule q waar in die toestand tot aan de toestand waarin p geldig is. Met de voorwaarde dat de eigenschap p niet gelabeled hoeft de worden aan een opvolgende state. De release-operator (R) is ook een binaire operator en wordt gebruikt om de tweede aan te geven eigendom geldt in alle staten langs een pad naar en inclusief de eerste staat die voldoet aan de eerste eigendom. De eerste eigenschap is niet vereist om uiteindelijk tevreden te zijn.

## 4.10 Fairness

Er wordt een fairness constraint opgelegd aan (de planner van) het systeem dat het proces eerlijk selecteert volgende worden uitgevoerd. Technisch gezien is een fairness constraint een voorwaarde voor uitvoeringen (paden) van het systeem model. Hoewel deze beperkingen geen eigenschappen zijn die moeten worden geverifieerd (het zijn eerder veronderstelde omstandigheden). worden afgedwongen door de implementatie), kunnen ze worden uitgedrukt in temporele logica.

fairness is geen te controleren eigenschap van een systeem. Denk eens aan de manier waarop asynchrone systemen zijn gemodelleerd: als een grafiek waarin elk knooppunt een mondiale systeemtoestand vertegenwoordigt, en elke opvolger van een knooppunt komt overeen met de nieuwe mondiale toestand die wordt bereikt door een bepaald van de processen die een lokale transitie maken. Zo'n model bevat doorgaans geen informatie over hoe vaak een proces wordt uitgevoerd, of hoe het volgende proces wordt uitgevoerd het uit te voeren proces wordt gekozen door de planner; in feite is het model onafhankelijk van de onderliggende besturingssysteem en dus van de planner: planning wordt niet-deterministisch behandeld. [?]

**Data variabelen** Dat variabelen zijn onder andere: water hoog en laag, en aanal schepen in de queue.

**Acties** Acties in het model zijn onder andere: invaren, uitvaren, deuren openen en sluiten, nivelleren

## 4.11 liveness properties

Veiligheidseigenschappen geven aan wat er wel of niet mag voorkomen, maar eis niet dat er ooit iets gebeurt. Levendigheidseigenschappen geven aan wat er moet gebeuren. De eenvoudigste vorm van een liveness-eigenschap garandeert dat er uiteindelijk iets goeds gebeurt. De 'goede zaak' vertegenwoordigt een wenselijke systeemtoestand, bijvoorbeeld de poorten open voor het wegverkeer. Een Booleaanse waarneembare nemen G : Tijd $\rightarrow$ 0, 1, waarbij G(t) $= 1$ drukt uit dat op tijdstip t de systeem in goede staat verkeert, kan deze levendigheidseigenschap tot uitdrukking worden gebracht volgens de formule: t $\in$ Tijd $^{\cdot}$ G(t). Met andere woorden: er bestaat een tijdstip waarop het systeem zich in de goede staat. Houd er rekening mee dat deze eigenschap niet in bounded kan worden vervalst tijd. Als voor enig tijdstip t0 alleen $\neg$G(t) is waargenomen t  t0, we kunnen niet klagen dat (1.2) uiteindelijk wordt geschonden zegt niet hoe lang het zal duren voordat de goede toestand zich zal voordoen. Een dergelijke liveness-eigenschap is niet sterk genoeg in de context van realtime systemen. Hier zou men graag een tijdgebonden zien wanneer de goede toestand ontstaat. Dit brengt ons bij het volgende soort vastgoed. begrensde responseigenschappen

[?, p.23]

Een propositie kan ook worden geschreven als : $AGEF_{[x,y]} \models \Phi_1$
Of geheel als functie:

$\forall$ b,e $\in$ Time $\bullet$ A(b,e) $=\int_b^e C(t)\,\mathrm{d}t \leq$ u(b,e)