Tinlab advanced algorithms eindopdracht

W. Oele

16 maart 2020

Ministerie van Infrastructuur
en Waterstaat

Den Haag, 16 maart 2020

Geachte specialist,

Na grondige analyse van het Nederlandse sluizenpark is gebleken dat renovatie van een groot aantal sluizen noodzakelijk is. Een eerste verkenning heeft ons geleerd dat het gecombineerd renoveren en automatiseren van het Nederlandse sluizenpark een aanzienlijke verbetering kan opleveren t.a.v.

- veiligheid
- efficiëntie
- capaciteit
- onderhoudskosten
- duurzaamheid

In het kader van het onlangs afgesloten klimaatakkoord heeft de Nederlandse overheid daarom besloten over te gaan tot een ingrijpende renovatie van de diverse sluizen die ons land rijk is. Op het ministerie van infrastructuur en waterstaat is helaas onvoldoende kennis van ict en systemen aanwezig om een en ander uit te voeren.

Wij vragen u een model (of een onderling samenhangend aantal modellen) aan te leveren, opdat ontwerpen van verschillende, volledig geautomatiseerde sluizen in de toekomst gerealiseerd kunnen worden.

Hoogachtend,

ds. A. M. B. ten Aar

1 De opdracht

Zoals gesteld in de brief is het de bedoeling dat een sluis gemodelleerd wordt en dat bewezen kan worden dat de te bouwen sluis een aantal eigenschappen bezit. Je krijgt daarbij veel vrijheid om zelf keuzes te maken. Daaruit volgt:

- dat de *onderbouwing* van die keuzes even belangrijk is als het uiteindelijke resultaat.
- dat er ook zaken mogen mislukken. Dat iets mislukt is niet erg, gesteld dat je duidelijk maakt *waarom* het mislukt is en wat geprobeerd is. Redenen van falen zijn doorgaans even interessant, zo niet interessanter, als de successen.

2 Beoordeling

De toetsing bestaat uit een totaalbeeld dat ontstaat uit:

- Het persoonlijke gedeelte van het verslag
- Het gezamenlijke verslag
- De mondelinge presentatie

2.1 Persoonlijk gedeelte

Bekeken wordt:

- Kwaliteit en diepgang van je verslag
- Gelezen artikelen en gebruikte bronnen
- Opbouw en structuur

Het door de docent aangeboden template hoeft niet strikt te worden gevolgt: Het is een suggestie waar je, naar eigen inzicht, van mag afwijken. Het gebruiken van LaTeX is uiteraard verplicht.

Het persoonlijke gedeelte kan beschouwd worden als een soort samenvatting van de leerstof. Belangrijkste vraag: Als je over een jaar veel bent vergeten en je je eigen verslag leest, kun je je de stof dan weer eigen maken? Levert je verslag een goede inleiding in het onderwerp verificatie en model checking?

2.2 Gezamenlijke gedeelte

Bij de beoordeling van het gezamenlijke gedeelte staan de volgende drie vragen centraal:

- Hoe zijn de wensen van de opdrachtgever geïnterpreteerd? Tot wat voor requirements/specificaties leiden deze? Anders gezegd: Wat betekent veilig, efficiënt, etc. en wat heb je aan bronnen geraadpleegd om tot een goede analyse te komen? (Dit laatste hoef je niet te beschrijven: het blijkt immers uit citaten of verwijzingen die je gebruikt.)
- Het model:
  - De modelcriteria van Vaandrager zijn op allerlei manieren tegenstrijdig. Welke keuzes en afwegingen heb je gemaakt en waarom?
  - Gemodelleerde onderdelen.
  - Werking van het model.
- Verificatie:
  - Wat heb je geverifieerd, waarom en hoe?
  - Als je iets *niet* kon verifiëren, waarom dan niet?
  - Een harde eis is *dat* er een aantal eigenschappen geverifieerd zijn. We modelleren een systeem immers middels Kripke structuren om harde uitspraken over eigenschappen van zo' n systeem te kunnen doen.

Enkele belangrijke opmerkingen:

- Sommige studenten denken dat ze bij het schrijven van verslagen per kilo worden betaald. . . Het is beslist niet de bedoeling om kilometers tekst te produceren. Beter een goed lopend, gestructureerd, korter document dan een waterig plofverslag.
- Voor het model geldt grofweg hetzelfde: Deze opdracht is niet een wedstrijdje complexiteit. Minimalisme/simplisme is, mits goed onderbouwd, beter dan een ingewikkeld vlaggeschip.
- Wat helpt bij het schrijven van een gestructureerd document is het van te voren neerzetten van een "skelet" met kernwoorden. Deze kernwoorden kun je uitsplitsen in begrippen die eronder vallen en zo krijg je, grofweg, de hoofdstukken, subhoofdstukken, etc. al op papier en kun je de structuur van je document beter in de gaten houden.

- Het is uiteraard toegestaan plaatjes in je verslag te zetten. Uppaal heeft de fijne eigenschap dat je een template kunt exporteren als pdf of png. Middels de `\includegraphics` functie kun je die in je LaTeX source verwerken.

- Wees met het in je verslag plaatsen van plaatjes zeer terughoudend. Een diagram/model/plaatje/tabel zet je alleen in de tekst als het *echt* iets toevoegt. In alle andere gevallen laat je het weg en verwerk je dat in de bijlagen.

2.3 Mondelinge presentatie

De mondelinge presentatie duurt 15 minuten *en niet meer!* In deze 15 minuten:

- presenteer je je model(len) en geef je kort en bondig een overzicht van de werking.

- laat je zien wat geverifieerd werd.

- beantwoord je wat vragen.

15 minuten is kort! Het lijkt misschien lang, maar als je eenmaal bezig bent, vliegt de tijd (docentenwijsheid;)). Dit betekent dat je goed moet nadenken over de vraag wat de grote lijn in je presentatie is. Diverse details *moet* je vanwege de tijd weglaten en daarmee is de mondelinge presentatie een mooie oefening in samenvatten: Wat is de essentie en wat laat je weg?

3 Tips en afwegingen

Daar deze opdracht erg vrij is, worden hier wat losse ideeën aangedragen. Dit zijn geen eisen en of ze op jouw model van toepassing zijn is ook nog maar de vraag. De aangedragen tips illustreren een denkwijze.

3.1 Analyse

Als je een sluis gaat modelleren is het uiteraard eerst zaak een goed beeld op te bouwen van de werking van een sluis. Stel je, voordat je het internet overhoop haalt, eerst eens wat relevante vragen:

- Zijn er verschillende soorten sluizen?

- Uit welke onderdelen bestaat een sluis?

- Hoe lang duurt het, voordat een boot door de sluis heen is?
- Wat voor stappen (states?) moeten doorlopen worden om een boot van de ene naar de andere kant te krijgen?
- Wat voor cijfers horen er bij het bovenstaande? Hoe lang duurt een bepaalde stap?
- etc.

Brainstormen en een "brainstormdocument" bijhouden heeft zijn voordelen.

3.2 Het model: algemene benadering

In de brief is te lezen dat het ministerie van infrastructuur en waterstaat een model wil hebben voor een sluis die volledig geautomatiseerd werkt. Dit betekent dat er ergens in je model een component moet zitten die, so to speak, de baas is. Die component is verantwoordelijk voor, bijvoorbeeld, het aan- en uitzetten van pompen, het openen en sluiten van sluisdeuren, etc. Deze "mastercontroller" deelt de lakens uit en communiceert met andere componenten.

3.3 Queues, integers en verificatie

Schepen kunnen gemodelleerd worden met simpele (bounded) integers die in een wachtrij geplaatst worden. Eén en ander betekent dat er ergens in je model een queue (of zelfs meerdere) geprogrammeerd moet worden. Wanneer we echter drie queues programmeren met een lengte van vijf, loopt bij het verifiëren van een eigenschap het aantal states al zo hoog op dat verifiëren onmogelijk is geworden...
Er zijn nu verschillende alternatieven:

- de queue lengte kleiner maken
- minder queues gebruiken
- queues helemaal niet gebruiken

Die laatste optie impliceert dat een simpele integer gebruikt wordt om het aantal schepen in de sluis bij te houden. Je bent nu niet meer in staat een individueel schip te "volgen", maar je hebt de state explosion wel enorm teruggebracht. Voor *beide* opties is iets te zeggen en het is daarom toegestaan om een model te veranderen of "uit te kleden", zodat een eigenschap die eerder niet verifieerbaar was, dat alsnog wordt...

Werken met meerdere, onderling samenhangende, modellen die gemaakt worden om er specifieke eigenschappen mee te verifiëren is een normale zaak en derhalve gewoon toegestaan. Let er, als je dit doet, wel op dat je goed documenteert wat er met welk model geverifieerd is.

3.4 Onverwachte omstandigheden

Wanneer een "echt" systeem gebouwd en in gebruik genomen wordt, kan er met een werkend systeem van alles misgaan:

- sensor gaat kapot
- menselijke fout
- mechanische fout
- ...

Je hoeft bij het modelleren geen rekening te houden met dit soort omstandigheden. Ga er van uit dat een gemodelleerd onderdeel doet wat het geacht wordt te doen.

3.5 Simplisme vs. realisme

Wanneer we in ons systeem met waterhoogte willen werken, zal die waarde ergens vandaan moeten komen. Het is realistisch om een sensor te modelleren die de waterhoogte "uitleest" en doorgeeft aan het systeem. Dit maakt het model realistischer, maar ook complexer.

Het doelbewust *niet* modelleren van een sensor is dan ook verdedigbaar: Met het verkregen simplisme ga je de state explosion tegen en dat maakt het verifiëren van eigenschappen mogelijk. Ook hier geldt dat je daar zelf keuzes in mag maken, gesteld dat je ze onderbouwt.

3.6 Liveness

Het verifiëren van liveness kan voor subtiele problemen zorgen. Deze komen met enige regelmaat voort uit wat we in de handleiding lezen:

The syntax $p \leadsto q$ denotes a leads to property meaning that whenever p holds eventually q will hold as well. Since Uppaal uses timed automata as the input model, this has to be interpreted not only over action transitions but also over delay transitions.

Anders gezegd: Als je in je model een state hebt zitten zonder invariant die het systeem vroeg of laat uit die state dwingt, kan het systeem eindeloos

in die state blijven hangen en zal liveness verificatie niet slagen. Het verhelpen daarvan kan een hoop werk opleveren...