

Project Formele Systeemmodellering voor Software: Specificatie en verificatie in TLA+/TLC

Groep Frank: Lukas De Loose, Ward De Muer,
Marie Vanzieleghem, Sean Deloddere

25 mei 2018

1 Inleiding

In dit verslag wordt uiteengezet op welke manier wij, de leden van groep Frank, trachtten het TLA+ project voor het vak Formele systeemmodellering voor Software uit te werken.

1.1 Veronderstellingen en vereenvoudigingen

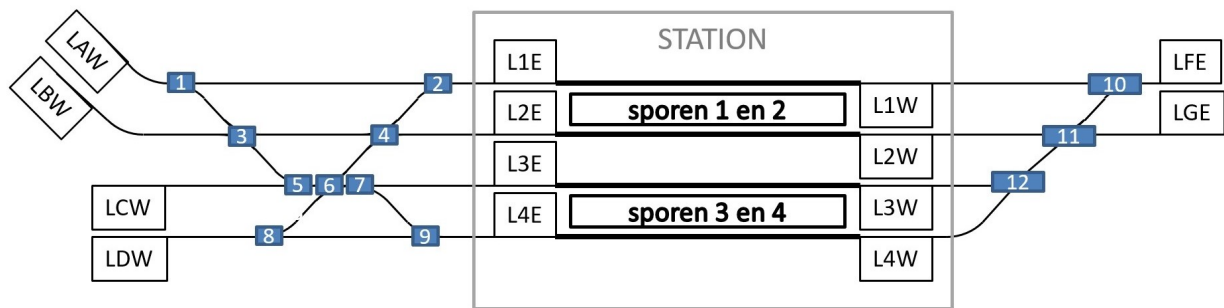
Hieronder worden de vereenvoudigingen en gemaakte veronderstellingen opgelijst. Als we vereenvoudigingen maakten, was dit doorgaans ofwel om de specificatie zo beknopt mogelijk, ofwel om de verificatie haalbaar te houden.

- Een trein die van het oosten komt, rijdt steeds naar het westen en vice versa. Een trein kan vanuit het station dus niet in dezelfde richting rijden als waar hij vandaan kwam.
- Treinen hebben geen specifieke bestemming. Ze nemen een bepaalde route als deze blijkt dat die vrij is.
- Lichten worden enkel groen als er een trein gedetecteerd werd. Dit om er voor te zorgen dat lichten niet onnodig groen worden en het vertrek van treinen aan andere lichten, die voor een rood licht staan wachten, belemmeren.
- We gaan ervan uit dat treinen elkaar kunnen passeren voor de buitenste lichten. Zo kan bijvoorbeeld een trein veilig van spoor 1 naar spoor F rijden, terwijl er een trein voor LFE staat te wachten.
- Als er aan één van de buitenste sporen een trein staat te wachten, wordt ervan uit gegaan dat er geen tweede trein op dat spoor zal toekomen.
- Er werd niet rechtstreeks rekening gehouden met gele lichten (zie later).
- De ordening van wissels kan u vinden in Figuur 1

1.2 Hulpmodules: Wissels, Sporen en Lichten

Er werd vanaf het begin van het project voor gekozen om de verschillende elementen van het treinsysteem te behandelen vanuit verschillende modules. Zo werden er modules aangemaakt om de wissels, de sporen, en de lichten in te stellen. In een overkoepelde stationmodule werd voor elk van de elementen van het treinsysteem een instantie aangemaakt.

De modules zoals hieronder beschreven zijn degene zoals ze in versie 2 van de code voorkomen. De module wissel heeft in versie 2 slechts 1 variabele bij te houden, en module licht zelfs in beide versies, waardoor het nut van de modules eigenlijk wegvalt, en de wissel- en lichtinstanties gemakkelijk door een simpele variabele in de



Figuur 1: De nummering van de wissels in het volledig model

overkoepelende stationmodule zouden kunnen vervangen worden. Voor eerdere versies (zie verder) hadden de verschillende modules echter soms meer en soms minder variabelen die moesten bijgehouden worden, waardoor we toch besloten de modules te behouden.

Wissels houden zoals gezegd 1 variabele bij, de 'bezet' variabele. Dit is een geheel getal dat aangeeft of de wissel gereserveerd werd door een passerende of een nog te passeren trein. Als een trein een wissel wil gebruiken of vrijgeven, incrementeert of decrementeert hij de 'bezet' variabele van de wissel via respectievelijk de 'reserveer' en 'vrij' functies in de wisselmodule. In versie 1 van de code is er ook nog een 'state' variabele in de wisselmodule opgenomen om de stand van de wissels aan te geven, maar deze werd weggelaten in versie 2 van de code. De reden hiervoor wordt verderop in het verslag nog duidelijk.

Sporen hebben ook een 'bezet' variabele, die een gelijkaardige werking heeft als bij wissels. Daarnaast heeft een spoor ook een 'trein' variabele. Deze geeft aan of er zich een trein op het spoor bevindt en wat het volgende spoor is waar deze trein naartoe zal rijden, óf wat de richting is waar de trein vandaan komt. Dit wordt aangegeven door een karakterstring die waarden in [N, 1, 2, 3, 4, A, B, C, D, F, G, W, E] kan aannemen, waarbij "N" aangeeft dat er geen trein is, "W" en "E" richtingen voorstellen en de andere waarden de verschillende sporen zijn. Bij sporen wordt de 'bezet' variabele aangepast door oproepen van de functies 'treinOp' en 'treinWeg'. De 'trein' variabele wordt gewijzigd via de functies 'zetTrein(richting)' en 'wegTrein'.

De twee variabelen in spoor hebben mogelijks op het eerste zicht schijnbaar dezelfde functie. Ze worden echter op een andere manier gebruikt. Het verschil tussen de twee is dat een trein via de 'bezet' variabele een spoor (of wissel, zie boven) bezet kan houden nog vóór dat de trein zich effectief op het spoor bevindt. Dit is om te voorkomen dat er meerdere treinen zich op hetzelfde moment naar hetzelfde spoor proberen begeven (of dezelfde wissel proberen gebruiken). De 'trein' variabele echter, wordt pas verschillend van "N" als de trein al op het spoor staat, en wordt gebruikt om de richting of de volgende bestemming van de trein bij te houden (in detail: zie verder). Wissels hebben géén 'trein' variabele omdat treinen niet kunnen stilstaan op wissels, maar zich in 1 beweging van het ene naar het andere spoor begeven.

Lichten ten slotte houden enkel hun kleur bij, dat "rood" of "groen" kan zijn en via de functies 'lichtRood' en 'lichtGroen' in de lichtmodule aangepast kan worden.

Er dient opgemerkt te worden dat voor treinen geen module werd geschreven, omdat een trein simpelweg kan voorgesteld worden door het aanpassen van een detectievariabele, zoals we in de module spoor doen. Als we hieronder spreken over treinen die zich verplaatsen, hebben we het dus over het doorgeven van deze detectie van de ene moduleinstantie naar de andere.

2 Opbouw en problemen

Voordat de uiteindelijke, verifieerbare specificatie bekomen werd, werden verschillende gemaakte beslissingen herzien en bepaalde aspecten toegevoegd of verwijderd. Er kunnen echter twee grote versies onderscheiden worden. Versie 1 behandelt het volledige station, waarbij aan alle veiligheidsvereisten voldaan is, maar de fairnessvereisten niet getest konden worden. Versie 2 voldoet aan alle vereisten, maar behandelt slechts de vereenvoudigde versie van het station.

Oorspronkelijk waren we begonnen met de 'trein' te verplaatsen van spoor naar wissel, wissel naar wissel

en wissel naar spoor. Dit door allemaal methoden te definiëren, zoals bijvoorbeeld `LCW_to_W08` die de trein van het licht LCW naar wissel 8 verplaatst, of `W08_to_W06` die de trein van wissel 8 naar wissel 6 verplaatst. Dit resulteerde echter in extreem veel variabelen, de module wissel moest namelijk ook nog een richting (cf. 'trein' variabele in spoor) bijhouden. We hadden ook nog veel meer functies die allemaal van elkaar afhingen en met teveel andere zaken rekening moesten houden. Kort gezegd, we maakten het onszelf veel te moeilijk dus zijn we met een schone lei begonnen.

In de secties hieronder bespreken we de werking van de nieuwe versies evenals de problemen die opdoken met de specificatie van de wissels en waarom we ervoor kozen om extra vereenvoudigingen door te voeren. Het leek ons nuttig deze evolutie in het verslag op te nemen aangezien er toch heel wat tijd in kroop om deze eerste versies te implementeren en dat ons dit een hoop nieuwe inzichten over het hele systeem opleverde, omtrent vereenvoudigingen en verificatie bijvoorbeeld, die we in de uiteindelijke versie konden benutten.

3 Versie 1: Gebruik van routes met wisselstanden

3.1 Routes

In deze versie wordt gebruik gemaakt van routes. Deze routes zijn telkens gedefinieerd van één van de buitenste sporen (sporen A t.e.m. F) naar één van de sporen in het station (sporen 1 t.e.m. 4) of omgekeerd. Zo is er bijvoorbeeld een route C1 die van spoor C naar spoor 1 gaat, of een route F2 die van spoor F naar spoor 2 gaat. Suboptimale routes, zoals bijvoorbeeld van spoor A naar spoor 1 via wissel 1, 3, 4 en 2, werden achterwege gelaten.

De routes zelf zijn in essentie een verzameling van voorwaarden (gebruikt als enabling condities) die aangeven welke wissels en sporen vrij moeten zijn om van het ene naar het andere spoor te gaan en in welke stand die wissels zich daarvoor moeten bevinden. Routes die in het station vertrekken bevatten enkel wissels, wat in overeenstemming is met de gemaakte veronderstelling dat bij wegrijden van het station geen rekening moet worden gehouden met treinen die voor lichten aan de buitenste sporen staan te wachten. Routes die van één van de buitenste sporen naar het station gaan, bevatten ook een spoor in het station dat vrij dient te zijn.

Routes kunnen worden gereserveerd of vrijgegeven. Reserveren van een route reserveert de wissels en eventueel het stationsspoor dat op de route ligt via de hierboven vermelde functies in de wissel- en spoormodules. Vrijgeven gebeurt analoog. Voor het reserveren en vrijgeven van de wissels in de routes werden hulpfuncties geschreven die 1 tot 4 wissels, meegegeven als argument aan een hulpfunctie, kunnen reserveren of vrijgeven. Dit om de code iets compacter te houden.

3.2 Next-acties

Mogelijke next-acties in de stationmodule zijn arriveren van een trein op één van de buitenste sporen, het op groen springen van een licht, het vertrekken van een trein naar zijn volgende spoor en het veranderen van de wisselstanden.

3.2.1 Arrive_X

Wanneer een trein arriveert op één van de buitenste sporen is aan de enabling conditie voldaan dat de 'trein' variabele van het spoor "N" was, wat wil zeggen dat er nog geen trein op het spoor stond te wachten. Dit is conform de hierboven besproken veronderstelling die zegt dat er geen trein zal arriveren op een buitenste spoor als er al één stond. Vervolgens worden de 'treinOp' en 'zetTrein' functies van het spoor opgeroepen. 'treinOp' wordt gebruikt om het spoor bezet te houden en te voorkomen dat er nieuwe treinen aankomen op hetzelfde spoor. De 'trein' variabele van de sporen wordt voor de westelijke buitenste sporen op "W" gezet, en voor de oostelijke op "E" door de 'zetTrein' functie. Hiermee wordt voor een spoor de richting aangegeven waaruit de trein zal komen.

3.2.2 LXY_groen

Het op groen springen van een licht kan enkel gebeuren wanneer er een trein op het spoor dat bij het licht hoort, staat te wachten, i.e. als de 'trein' variabele van het spoor verschillend is van "N". Meer specifiek zal deze variabele de waarde "W" en "E" aannemen voor respectievelijk lichten in westelijke richting en lichten in oostelijke richting. Dit is vooral nuttig voor treinen die op sporen in het midden van het station staan, om te kunnen voldoen aan de gemaakte veronderstelling dat treinen enkel van het westen naar het oosten of van het oosten naar het westen doorheen het station kunnen rijden. Het op groen zetten van het licht gaat gepaard met het zoeken naar een beschikbare route. Als er een route gevonden werd die vrij is, wordt de route gereserveerd en krijgt de 'trein' variabele van het vertrekspoor het eindspoor van de route mee. De 'trein' variabele van het vertrekspoor heeft nu dus één van de waarden in [1, 2, 3, 4, C, D, F, G]. Op deze manier kan men achterhalen welke route moet worden vrijgegeven bij het aankomen van de trein op het eindspoor.

3.2.3 LXY_vertrek

Een nodige voorwaarde voor het vertrekken van een trein is uiteraard het niet-rood zijn van het licht waar de trein aan staat te wachten. Als aan deze voorwaarde voldaan is, wordt gekeken wat de waarde van de treinvariabele op het vertrekspoor is. Deze variabele houdt nu namelijk bij, aangezien het licht groen is, waar de trein naartoe moet. Naast het te weten komen welke route moet worden vrijgegeven, wordt de treinvariabele bij het vertrekspoor ook gebruikt om de trein op zijn correcte bestemming te laten toekomen.

3.2.4 Change_WX

Wissels kunnen op elk gewenst moment veranderen van toestand.

3.3 Veiligheids- en fairnessvereisten

Volgens de TypeInvariant voor de 'bezet' variabele in spoor en wissel mag deze enkel 0 of 1 zijn. Als deze variabele de waarde 2 of meer aanneemt, dan zijn er meerdere treinen die dezelfde wissel of hetzelfde spoor bezet houden, waardoor er dus een botsing optreedt. De modelchecker leert ons dat de TypeInvariant niet geschonden wordt, en er dus voldaan is aan de veiligheidsvereisten (zie figuur 2).

```
Model checking completed. No error has been found.
  Estimates of the probability that TLC did not check all reachable states
  because two distinct states had the same fingerprint:
    calculated (optimistic): val = .0054
    based on the actual fingerprints: val = 1.5E-4
1250627473 states generated, 85436856 distinct states found, 0 states left on queue.
The depth of the complete state graph search is 34.
The average outdegree of the complete state graph is 1 (minimum is 0, the maximum 18 and the 95th percentile is 3).
Finished in 02h 25min at (2018-05-25 18:34:45)
```

Figuur 2: Resultaat van de veiligheidsvereisten bij versie 1

De fairnessvereisten waren iets gecompliceerder. Wanneer er hier geen voorwaarden op worden gelegd, blijven er treinen eendeloos aan een rood licht staan. Om dit te verifiëren definiëren we de property *Detectie-Groen*, die zegt dat als er een trein staat te wachten en het licht is rood, dit licht uiteindelijk groen wordt. Zoals verwacht werd deze property geschaad. Om dit op te lossen voeren we Strong Fairness in voor zowel het groen worden van een licht, en het vertrekken van een trein. Ook voeren we Weak Fairness in voor het veranderen van de toestand van de wissels.

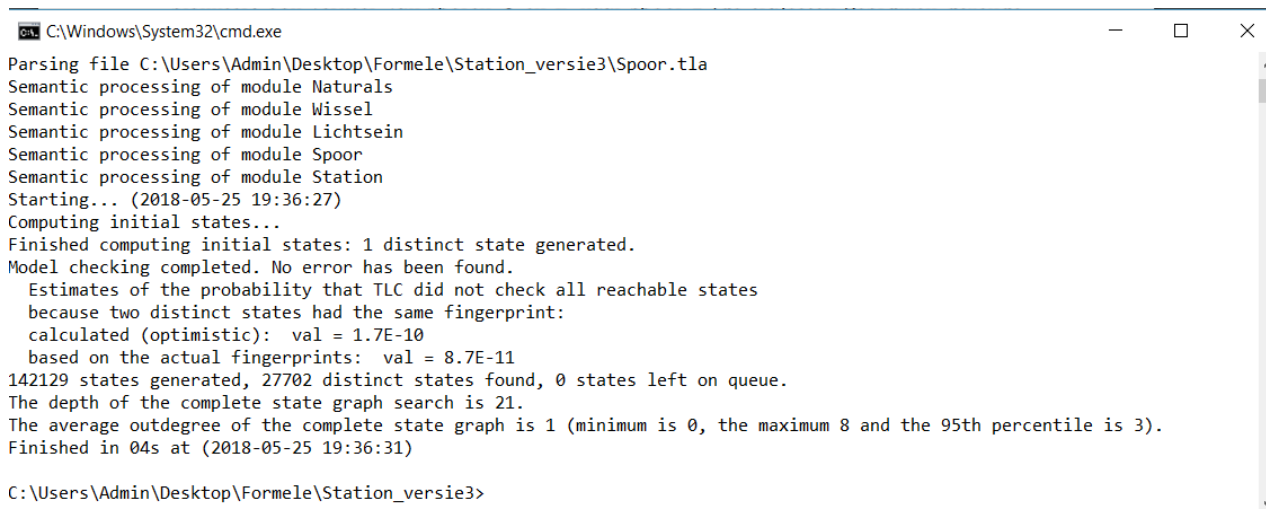
Het willekeurig veranderen van de wissels zorgt echter voor extreem veel states, waardoor de JVM een out-of-heap memory genereerde (zelfs voor 8G). Hierdoor kon voor deze versie dus nooit getest worden of ook voldaan werd aan de fairnessvereisten met het invoeren van de Strong en Weak Fairness. Voor deze versie vervuldigd werd met de sporen A en B, werd de fairnessvereiste toen ook al eens getest. Omwille van het lange runnen en de tijdsdruk hebben we deze test nooit kunnen laten aflopen en diende er een vereenvoudigd model gezocht te worden

4 Versie 2: Weglaten van wisselstanden

Om het aantal toestanden te verlagen, werd ervoor gekozen om de wissels niet meer van toestand te laten veranderen. De 'state' variabele van de wissels werd dus weggelaten.¹ Er wordt in de plaats daarvan verondersteld dat wanneer een trein een route reserveert, de wissels automatisch juist worden gezet (wat geen invloed heeft op de vereisten van het systeem). Om deze aanpassing door te voeren moesten enkel de routes in de code aangepast worden, om ervoor te zorgen dat de stand van de wissels geen deel meer uitmaakt van de enabling condities voor het reserveren van een route.

4.1 Veiligheids- en fairnessvereisten

Zoals verwacht geven de veiligheidsvereisten van deze vereenvoudiging geen problemen (zie figuur 3).



```
C:\Windows\System32\cmd.exe
Parsing file C:\Users\Admin\Desktop\Formele\Station_versie3\Spoor.tla
Semantic processing of module Naturals
Semantic processing of module Wissel
Semantic processing of module Lichtsein
Semantic processing of module Spoor
Semantic processing of module Station
Starting... (2018-05-25 19:36:27)
Computing initial states...
Finished computing initial states: 1 distinct state generated.
Model checking completed. No error has been found.
  Estimates of the probability that TLC did not check all reachable states
  because two distinct states had the same fingerprint:
  calculated (optimistic):  val = 1.7E-10
  based on the actual fingerprints:  val = 8.7E-11
142129 states generated, 27702 distinct states found, 0 states left on queue.
The depth of the complete state graph search is 21.
The average outdegree of the complete state graph is 1 (minimum is 0, the maximum 8 and the 95th percentile is 3).
Finished in 04s at (2018-05-25 19:36:31)
C:\Users\Admin\Desktop\Formele\Station_versie3>
```

Figuur 3: Resultaat van de veiligheidsvereisten bij versie 2

We definiëren dezelfde property *DetectieGroen*. Om te voldoen aan deze property voeren we Strong Fairness in voor de **LXY_groen** en Weak Fairness in voor **LXY_vertrek**. Zoals verwacht wordt de property niet geschonden, en is er dus zowel aan de fairness als aan de veiligheidsvoorwaarden voldaan (zie Figuur 3)².

5 Conclusie over de bekomen resultaten

We zijn er in geslaagd om, volgens onze vereenvoudigingen en veronderstellingen, het hele station te modeleren waarbij met zekerheid voldaan is aan de veiligheidsvereisten. De fairnessvereisten zijn voldaan voor het vereenvoudigde model, en deze kunnen zonder problemen uitgebreid worden naar het volledige model (doch met een pak extra rekentijd). De enige tekortkoming is het ontbreken van gele lichten. De groene lichten kunnen echter beschouwd worden als gele lichten, aangezien het volgende (bereikbare) licht altijd rood is, en dus toch aan de deze voorwaarde voldaan is.

We kunnen concluderen dat ons systeem, ondanks lange rekentijden, aan alle specificaties en vereisten voldoet.

¹Door deze aanpassing vervalt het nut van de aparte wisselmodule, deze werd echter behouden zodat de volledige code niet behoefde aangepast te worden

²Om de rekentijden te beperken werd er gebruik gemaakt van het vereenvoudigd stationmodel

Model Checking Results



General

Start time:	Fri May 25 16:12:32 CEST 2018
End time:	Fri May 25 19:06:25 CEST 2018
TLC mode:	Breadth-first search
Last checkpoint time:	
Current status:	Not running
Errors detected:	<u>No errors</u>
Fingerprint collision probability:	calculated: 1.7E-10, observed: 8.7E-11

Statistics

State space progress (click column header for graph)

Time	Diam...	States Found	Distinct Sta	Queue Siz
2018-05-25 19:...	21	142129	27702	0
2018-05-25 16:...	17	126921	26177	1892
2018-05-25 16:...	8	3125	1171	610
< [Progress Bar] >				

Coverage at 2018-05-25 19:06:07

Module	Location	Count	
Wissel	line 12, col 14 to line 12, c...	87318	^ v
Spoor	line 13, col 15 to line 13, c...	63288	
Wissel	line 13, col 9 to line 13, col	87318	
Station	line 14, col 106 to line 14, ...	47520	

Figuur 4: Resultaat van versie 2, met DetectieGroen property (in TLA+ Toolbox)