Tinlab advanced algorithms

W. Oele

17 februari 2019

Deze les

- state transition diagrams
- state en state explosion
- invarianten en guards
- tijd
- Zeno gedrag en deadlock

Overzicht

- Een te bouwen systeem (system to be) modelleren we middels een *state transition diagram*.
- Met temporele logica verifiëren we of een uit requirements en specificaties voortkomende uitspraak geldig is voor ons model.

Modelleren

We modelleren een te bouwen systeem vanuit de gedachte dat een systeem:

- zich kan bevinden in een aantal toestanden of states.
- "werkt" door van de ene toestand over te gaan in de andere.

State transition diagrams

Er bestaan veel varianten van state transition diagrams:

- state transition diagrams
- labeled state transition diagrams
- timed state transition diagrams
- labeled timed state transition diagrams
- input-output state transition diagrams
- input enabled input-output state transition diagrams
- Kripke structuren
- ...

Boek en Uppaal

- In het boek wordt uitgegaan van Kripke structuren.
- Uppaal werkt met labeled timed state transition diagrams.

Kripke structuren: states en transities

- Een *state*:
  - is een beschrijving van het systeem gedurende een bepaald tijdsinterval.
  - bevat de waarden van alle variabelen van het systeem gedurende dat interval.
  - is *niet* een moment in de tijd.

Kripke structuren: states en transities

- Een *state*:
  - is een beschrijving van het systeem gedurende een bepaald tijdsinterval.
  - bevat de waarden van alle variabelen van het systeem gedurende dat interval.
  - is *niet* een moment in de tijd.
- Een "werkend" systeem kan men modelleren middels het doorlopen van een aantal toestanden.

Kripke structuren: states en transities

- Een *state*:
  - is een beschrijving van het systeem gedurende een bepaald tijdsinterval.
  - bevat de waarden van alle variabelen van het systeem gedurende dat interval.
  - is *niet* een moment in de tijd.
- Een "werkend" systeem kan men modelleren middels het doorlopen van een aantal toestanden.
- Een "overgang" van de ene toestand naar de andere noemt men een *transitie*.

Voorbeelden

Welke states heeft een stoplicht?

Voorbeelden

Welke states heeft een stoplicht?

- rood
- oranje
- groen

Voorbeelden

Welke states heeft een stoplicht?

- rood
- oranje
- groen

Welke states heeft een magnetron?

Voorbeelden

Welke states heeft een stoplicht?

- rood
- oranje
- groen

Welke states heeft een magnetron?

- aan
- uit
- deur open
- deur dicht
- ...

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?
- een systeem met 2 lampjes?

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?
- een systeem met 2 lampjes?
- een systeem met 5 lampjes?

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?
- een systeem met 2 lampjes?
- een systeem met 5 lampjes?
- een systeem met 10 booleans?

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?
- een systeem met 2 lampjes?
- een systeem met 5 lampjes?
- een systeem met 10 booleans?
- een systeem met 3 tuimelschakelaars, twee lampjes en een integer?

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?
- een systeem met 2 lampjes?
- een systeem met 5 lampjes?
- een systeem met 10 booleans?
- een systeem met 3 tuimelschakelaars, twee lampjes en een integer?
- een systeem met 2 doubles?

Oefenvragen

Hoeveel states heeft:

- een systeem met 1 lampje dat aan of uit kan?
- een systeem met 2 lampjes?
- een systeem met 5 lampjes?
- een systeem met 10 booleans?
- een systeem met 3 tuimelschakelaars, twee lampjes en een integer?
- een systeem met 2 doubles?
- stoplichten op het kruispunt Rochussenstraat/'s Gravendijkwal?

Het state explosion problem

- Het aantal toestanden, waarin een systeem zich kan bevinden wordt bij ingewikkelde systemen al snel te groot om goed te kunnen modelleren.
- Exponentiële groei van het aantal toestanden vormt een belangrijke beperking van model checking.

Uitdagingen:

- Hoe beperken we het aantal toestanden?
- Hoe kunnen we efficiëntere algoritmen vinden, zodat complexere systemen te verifiëren zijn?

Modelleren met Kripke structuren

Voor het modelleren van een systeem hebben we nodig:

- alle states van het systeem. We stoppen deze in een verzameling:
  - S: de verzameling van alle states van het systeem.
  - Elke individuele state noemen we $s_0, s_1, s_2, ...s_n$

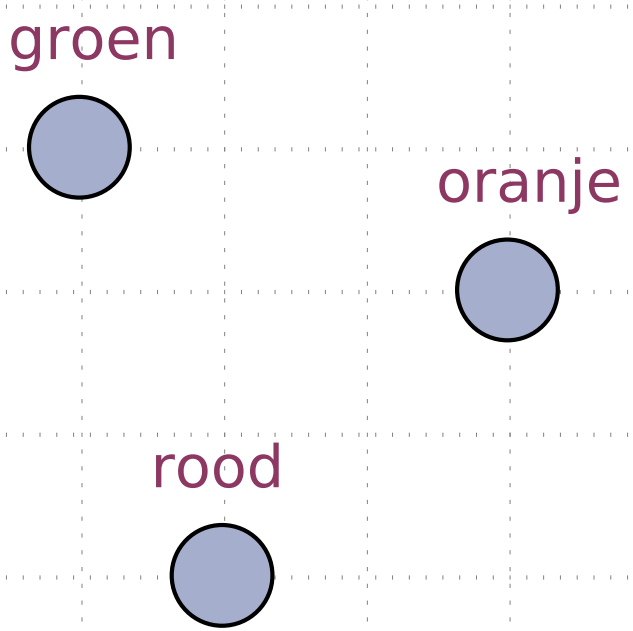
Kortom:

$$s_0..s_n \in S$$

Ons model is een tuple met daarin de verzameling states:

$$M = (S)$$

Voorbeeld: het stoplicht



De initial state

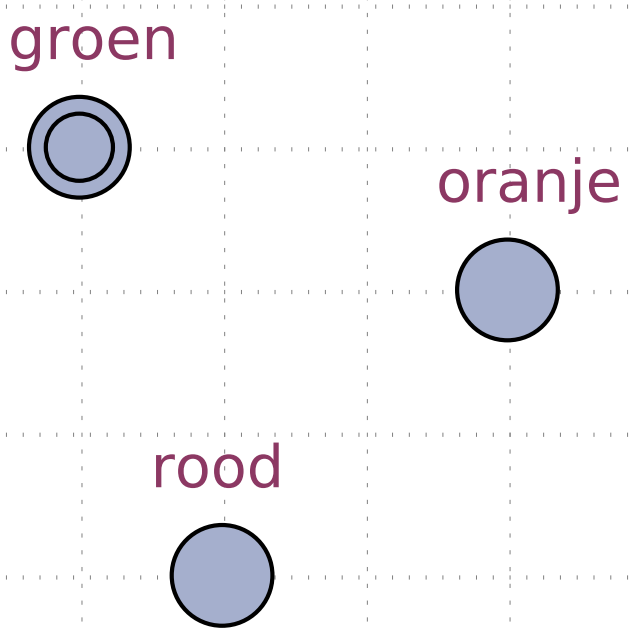
- Wanneer we een systeem in werking zetten, bevindt dat systeem zich in een begintoestand: de *initial state*.
- De initial state is één van de states uit S.
- Er kunnen meer initial states zijn, maar doorgaans is het er één.

In ons model zit derhalve:

- S
- $S_0 \subseteq S$

$$M = (S, S_0)$$

Voorbeeld: het stoplicht



Transities

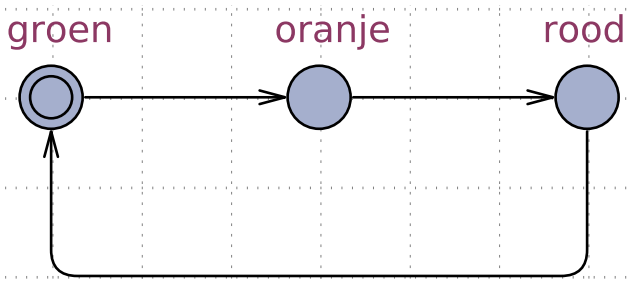
Transities tussen states vormen een relatie

$$R \subseteq S \times S$$

Ons stoplicht:

$$S = \{groen, oranje, rood\}$$
$$R \subseteq S \times S =$$
$$\{(groen, groen), (groen, oranje), (groen, rood),$$
$$(oranje, groen), (oranje, oranje), (oranje, rood),$$
$$(rood, groen), (rood, oranje), (rood, rood)\}$$

Voorbeeld: het stoplicht



$$R = \{(groen, oranje), (oranje, rood), (rood, groen)\}$$

Ons model is nu:

$$M = (S, S_0, R)$$

Transities

- De systemen die wij modelleren zijn *reactief*: Systemen kunnen eindeloos "rondjes lopen" door een aantal toestanden.
- Belangrijk gevolg: Voor elke state $s \in S$ geldt dat er een state s' bestaat zodanig dat geldt: $R(s, s')$
- *"Elke state heeft een uitgaande transitie."*

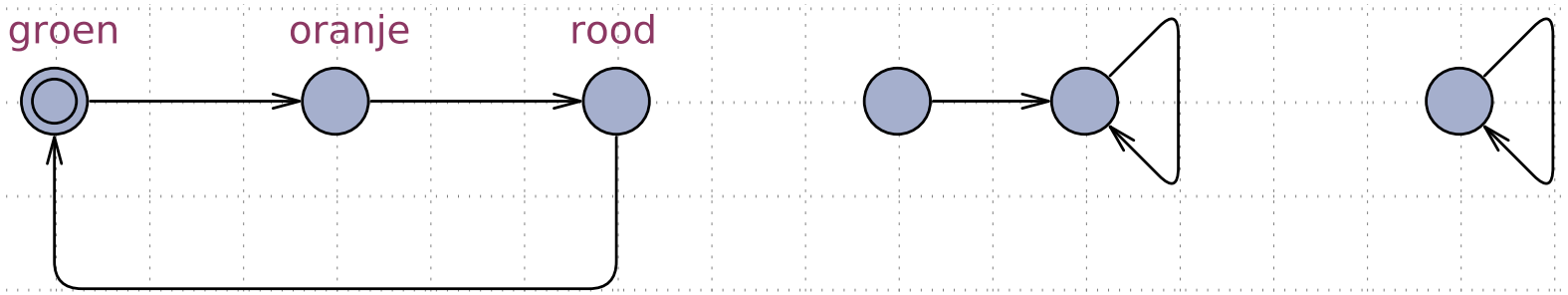
Transities

- De systemen die wij modelleren zijn *reactief*: Systemen kunnen eindeloos "rondjes lopen" door een aantal toestanden.
- Belangrijk gevolg: Voor elke state $s \in S$ geldt dat er een state s' bestaat zodanig dat geldt: $R(s, s')$
- *"Elke state heeft een uitgaande transitie."*

Een transitierelatie, waarin elke state een uitgaande transitie heeft noemt men *totaal*.

- Alle transitierelaties in de systemen die wij modelleren zijn totaal.

Voorbeelden



Labels

Om uitspraken te kunnen doen over ons systeem gebruiken we:

- Een verzameling atomaire proposities (AP): proposities die niet verder op te delen zijn in kleinere/kortere proposities
- Een *labeling functie: L*

De labeling functie is een functie die elke state "labeled" met een verzameling atomaire proposities die waar zijn in die state:

$$L = S \rightarrow 2^{AP}$$

Op de precieze rol van labels in Kripke structuren wordt later bij het verifiëren teruggekomen.

De Kripke structuur

Een Kripke structuur bestaat uit een 4-tuple $M = (S, S_0, R, L)$ met daarin:

- S: de verzameling van alle states van dit systeem.
- $S_0 \subseteq S$: de verzameling van beginstates.
- $R \subseteq S \times S$: de transitierelatie.
- $L : s \rightarrow 2^{AP}$: labels, waarmee we iedere state labelen met atomaire proposities die waar zijn in die state.

Invarianten en guards

- Invarianten en guards zijn condities.

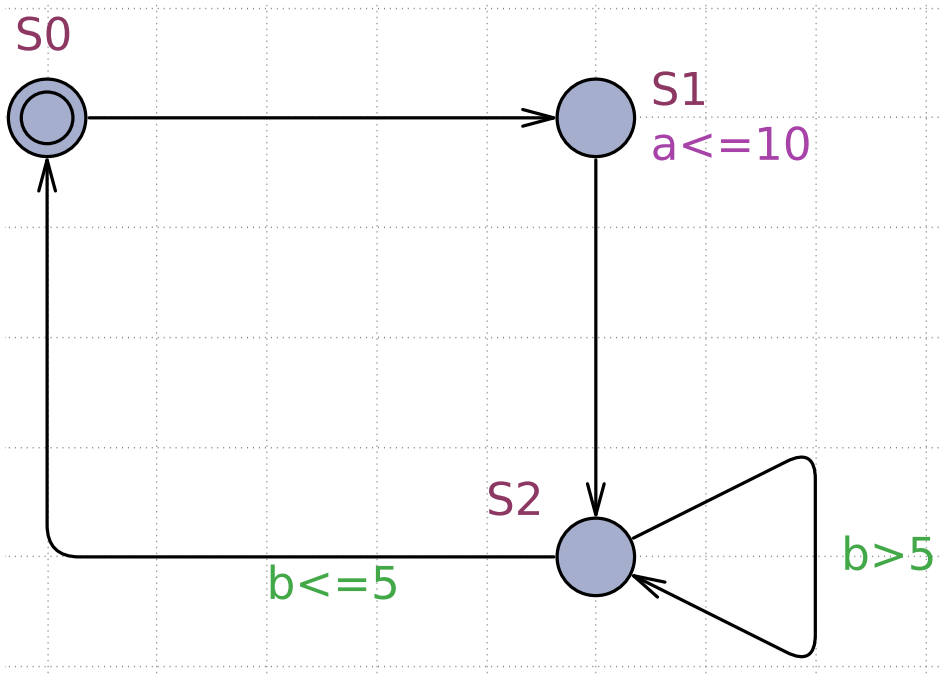
Invarianten en guards

- Invarianten en guards zijn condities.
- Een *invariant* is een conditie die geldt in een state en die *altijd* waar is wanneer de automaat zich in die state bevindt.

Invarianten en guards

- Invarianten en guards zijn condities.
- Een *invariant* is een conditie die geldt in een state en die *altijd* waar is wanneer de automaat zich in die state bevindt.
- Een *guard* is een conditie die geldt in een transitie. M.a.w. De transitie kan alleen genomen worden wanneer de guard geldig is.

Invarianten en guards: voorbeeld



Tijd

- Tijd wordt in Uppaal beschouwd als een continu verschijnsel.
- Tijd wordt in Uppaal bijgehouden met klokken.
- Alle klokken in het model lopen even snel (no Einstein allowed).
- Klokwaarden kunnen worden uitgelezen en worden verwerkt in invarianten en guards.
- Klokken kan men resetten in transities.

Tijd, klokken, states en transities

Enkele losse, doch belangrijke, opmerkingen:

- Tijd verstrijkt *uitsluitend* in states.
- De hoeveelheid tijd die verstrijkt "tijdens" een transitie is 0.
- Klokken worden altijd gereset naar 0.

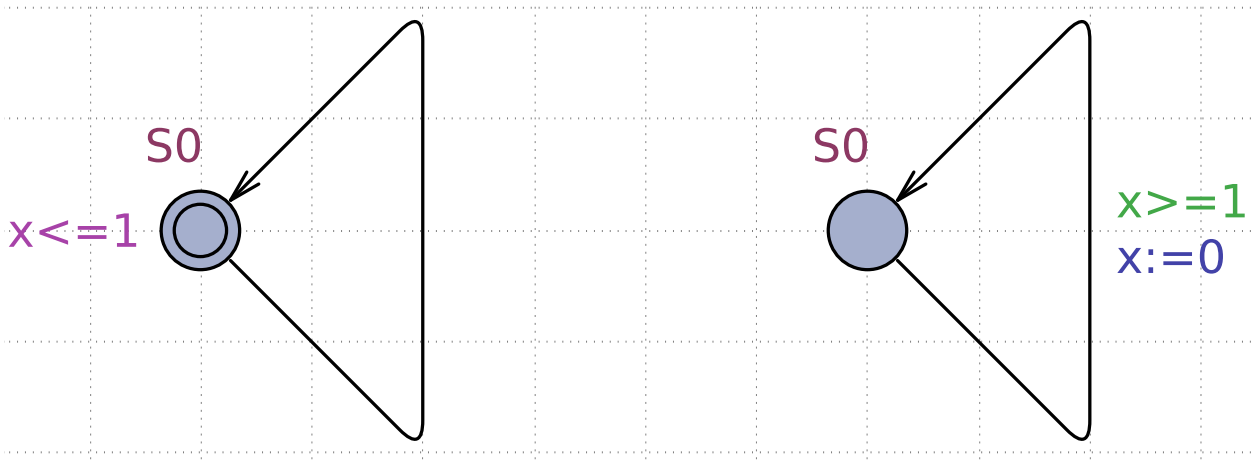
Zeno gedrag

- Zeno gedrag is genoemd naar de Griekse filosoof Zeno van Elea.
- Zie: `https://en.wikipedia.org/wiki/Zeno_of_Elea`.
- Zeno is bekend vanwege diverse paradoxen.
- Voor ons is van belang de paradox van Achilles en de schildpad.
- Zie: `https://en.wikipedia.org/wiki/Zeno%27s_paradoxes#Achilles_and_the_tortoise`

Zeno gedrag

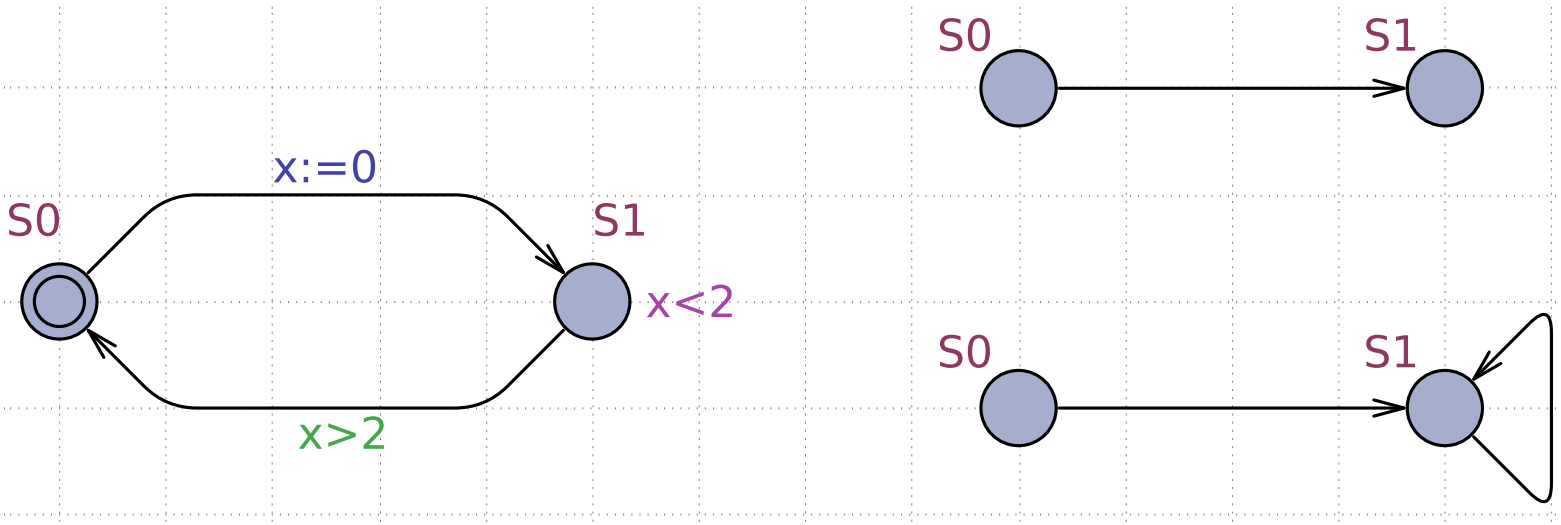
Zeno gedrag:

- Probleem: de mogelijkheid dat in een eindige hoeveelheid tijd een oneindig aantal handelingen kan worden verricht.



Deadlock

Deadlock ontstaat wanneer een combinatie van invarianten en guards het verstrijken van tijd verhindert.



Deadlock en Zeno gedrag

- Zowel Zeno gedrag als deadlock zijn soms erg moeilijk op te merken in een model.
- In Uppaal kan in de verifier de volgende query worden ingevoerd:

  `A[] not deadlock`
- Deze (later uit te leggen) expressie bekijkt of er in je model een deadlock aanwezig is.

Oefening

Modelleer de Fenix pd32 zaklamp:

Model 1 Simpel aan en uit

Oefening

Modelleer de Fenix pd32 zaklamp:

- Model 1 Simpel aan en uit
- Model 2 Model 1 + eco, low, mid, high en turbo stand

Oefening

Modelleer de Fenix pd32 zaklamp:

- Model 1 Simpel aan en uit
- Model 2 Model 1 + eco, low, mid, high en turbo stand
- Model 3 Model 2 + geheugenfunctie

Oefening

Modelleer de Fenix pd32 zaklamp:

- Model 1 Simpel aan en uit
- Model 2 Model 1 + eco, low, mid, high en turbo stand
- Model 3 Model 2 + geheugenfunctie
- Model 4 Model 3 + oververhittingsbeveiliging

Oefening

Modelleer de Fenix pd32 zaklamp:

- Model 1 Simpel aan en uit
- Model 2 Model 1 + eco, low, mid, high en turbo stand
- Model 3 Model 2 + geheugenfunctie
- Model 4 Model 3 + oververhittingsbeveiliging
- Model 5 Model 4 + Aan/uit- en modeknop (non Zeno!)

Oefening

Modelleer de Fenix pd32 zaklamp:

- Model 1 Simpel aan en uit
- Model 2 Model 1 + eco, low, mid, high en turbo stand
- Model 3 Model 2 + geheugenfunctie
- Model 4 Model 3 + oververhittingsbeveiliging
- Model 5 Model 4 + Aan/uit- en modeknop (non Zeno!)
- Model 6 Model 5 + strobe en sos functie

Uiteraard zijn alle modellen deadlock en Zeno vrij. . .