Wat is Model Checking?

Frits Vaandrager

24 maart 2008 (kleine aanpassing 9 november 2010)

Samenvatting

Afgelopen maand werd bekend dat de ACM Turing Award dit jaar uitgereikt zal worden aan Ed Clarke, Allen Emerson en Joseph Sifakis voor hun werk op het gebied van "model checking". De Turing Award, vaak aangeduid als de Nobelprijs voor de Informatica, is vernoemd naar de Britse wetenschapper Alan Turing, één van de pioniers van de Informatica. In deze bijdrage leg ik uit wat model checking is en hoe het werkt.

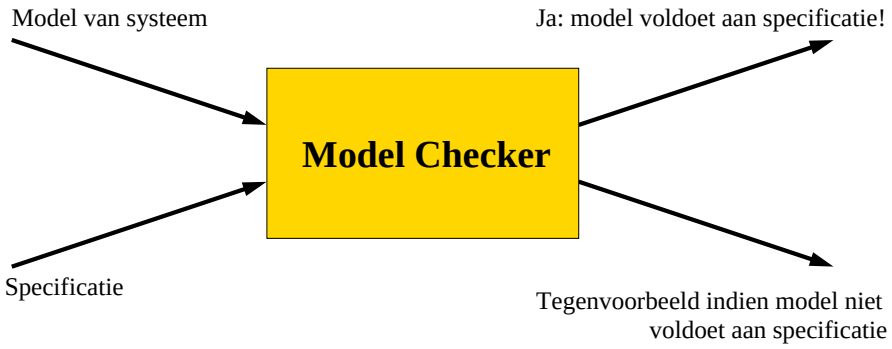
Achtergrond

Computers zijn beslist de meest complexe dingen die de mens ooit geconstrueerd heeft. De complexiteit van computertechnologie wordt door veel mensen schromelijk onderschat en bijna dagelijks staan er dan ook berichten in de krant over problemen die veroorzaakt zijn door fouten met computers. Zo hoorden we de laatste weken over gekraakte toegangspasjes en OV-dagkaarten, storingen in de nieuwe Roertunnel, 730.000 verloren belastingaangiften en een storing bij het internetbankieren van de Rabobank.

Figuur 1: Model checking pioniers Clarke, Emerson en Sifakis.

Bedrijven die computersystemen bouwen willen fouten natuurlijk het liefst zo vroeg mogelijk vinden, bij voorkeur voordat de systemen in gebruik worden genomen.

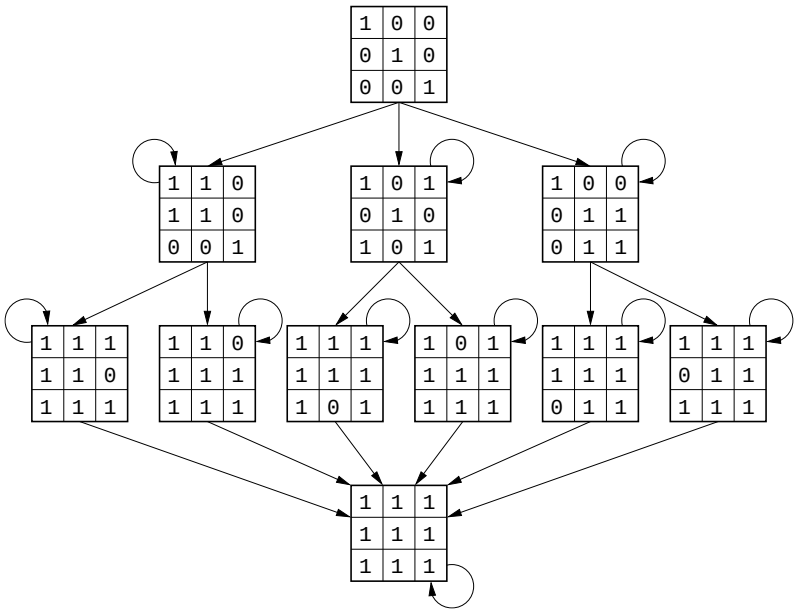
Model checking, voor het eerst beschreven in artikelen van de laureaten Clarke, Emerson en Sifakis, is een succesvolle methode om te bewijzen dat een ontwerp van een computersysteem voldoet aan de eisen die er aan gesteld worden. Het basisidee staat weergegeven in Figuur 2. Een *model checker* is een computerprogramma met als invoer een ontwerp van een systeem (het *"model"*) en een eigenschap (de *"specificatie"*) waaraan het systeem moet voldoen. De model checker berekent dan of het model voldoet aan de specificatie. Indien dit niet het geval is dan geeft het programma een tegenvoorbeeld dat laat zien waarom het model niet aan zijn specificatie voldoet. Door dit tegenvoorbeeld te bestuderen kun je er achter komen wat nu precies de oorzaak van het probleem is. Na model en/of specificatie verbeterd te hebben kun je het dan nog eens proberen. Het idee is dat door een groot aantal eigenschappen van het model te checken, het vertrouwen in de correctheid van het systeemontwerp toeneemt.



Figuur 2: De model checking aanpak.

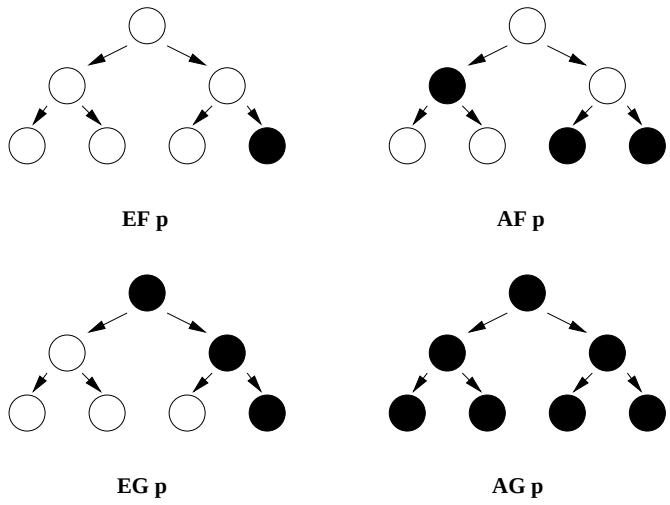
Modellen

Model checkers richten zich met name op modellen waarin *dynamisch gedrag* van een systeem beschreven wordt: in welke toestanden kan het systeem zich bevinden en welke overgangen tussen toestanden zijn er mogelijk? Om het gebruik van model checkers te illustreren kijken we naar een vraag die ooit gebruikt is bij de Nationale Wetenschapsquiz: "Zes vriendinnen hebben ieder één roddel. Ze bellen elkaar. In elk gesprek wisselen ze alle roddels uit



Figuur 3: Toestandsruimte voor 3 roddelende vriendinnen.

die ze op dat moment kennen. Hoeveel gesprekken zijn er minstens nodig om iedereen op de hoogte te brengen van alle zes de roddels?" Figuur 3 laat de toestandsruimte zien voor de vereenvoudigde versie met 3 vriendinnen. Een toestand bestaat uit een 3 bij 3 matrix waarin voor iedere vriendin wordt bijgehouden welke roddels ze weet: indien vriendin i de roddel van vriendin j weet schrijven we een 1 in het vakje in rij i en kolom j en als ze hem niet weet schrijven we een 0. In de begintoestand, bovenaan in het diagram, weet iedere vriendin alleen haar eigen roddel en dus staan er 1-tjes op de diagonaal en 0-en elders. Toestandsovergangen vinden plaats wanneer vriendinnen elkaar bellen. In de begintoestand zijn er drie mogelijke overgangen: vriendinnen 1 en 2 bellen elkaar, vriendinnen 1 en 3, of vriendinnen 2 en 3. In de nieuwe toestanden worden de rijen die horen bij de telefonerende vriendinnen aangepast: alle roddels worden uitgewisseld. Niet ieder telefoongesprek leidt noodzakelijk tot een nieuwe toestand: soms levert een gesprek geen nieuwe informatie op, dit correspondeert met een lusje van de toestand naar zichzelf. In totaal zijn er 11 bereikbare toestanden.



Figuur 4: Operatoren van de temporele logica CTL.

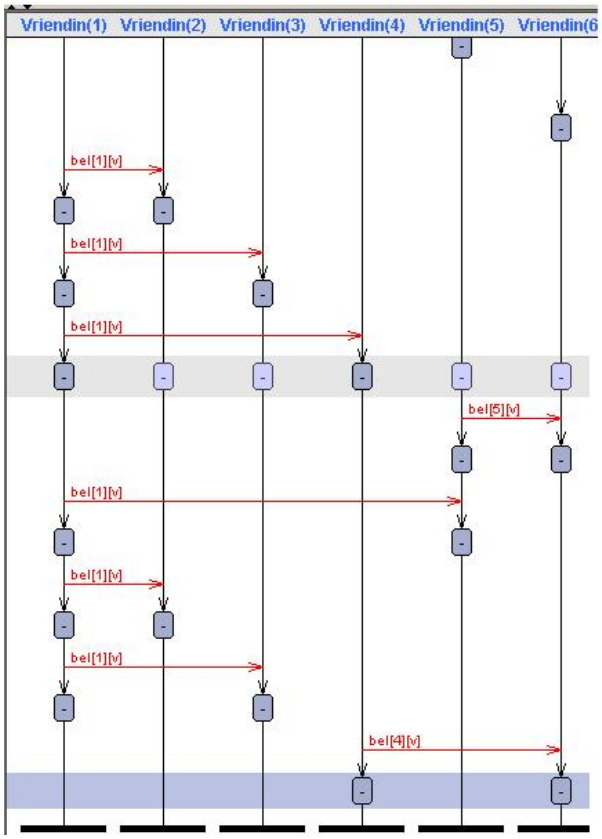
Specificaties

Eigenschappen van modellen kunnen op een eenvoudige wijze beschreven worden in de taal van de *temporele logica*. Figuur 4 illustreert enkele operatoren uit de temporele logica CTL, die bedacht is door Clarke en Emerson. In CTL betekent **E** "Er bestaat een pad waarvoor geldt dat", **A** "Voor alle paden geldt dat", **F** "uiteindelijk" en **G** "altijd". Stel dat **p** een eigenschap is van toestanden, bijvoorbeeld "vriendin 2 weet de roddel van vriendin 1", en stel dat **s** een specifieke toestand is. In **s** geldt

- **EF p** indien er een pad is van **s** naar een toestand waarin **p** geldt,
- **AF p** indien op alle uitgaande paden van **s** een toestand ligt waarin **p** geldt,
- **EG p** indien er een pad vertrekt in **s** met uitsluitend toestanden waar **p** geldt,
- **AG p** indien op alle uitgaande paden vanuit **s** overal **p** geldt.

Een voorbeeld van een formule die geldt in de begintoestand van ons voorbeeld is **AG** "iedere vriendin weet haar eigen roddel". Een voorbeeld van een formule die niet geldt is **AG** "als i de roddel van j weet dan weet j de roddel van i". De vraag uit de wetenschapsquiz kunnen we oplossen door een

model checker te vragen naar het kortste pad dat aantoont dat eigenschap **EF** "iedere vriendin weet iedere roddel" geldt in de begintoestand. In het geval met 3 vriendinnen volgt direct uit Figuur 3 dat dit kortste pad lengte 3 heeft. Voor 6 vriendinnen, de instantie uit de wetenschapsquiz, bevat het toestandsdiagram vele duizenden toestanden en kan een oplossing alleen gevonden worden door nadenken of brute rekenkracht. Figuur 5 toont een minimale oplossing met 8 gesprekken die gevonden is door de model checker Uppaal. Wiskundigen hebben bewezen dat n vriendinnen altijd minimaal $2n - 4$ gesprekken nodig hebben om alle roddels uit te wisselen.[1]



Figuur 5: Door model checker gevonden oplossing voor 6 vriendinnen.

[1] C.A.J. Hurkens. Spreading gossip efficiently. *Nieuw Archief voor Wiskunde*, 5/1(2):208–210, Juni 2000.

Toestandsexplosies

Wanneer het aantal vriendinnen toeneemt dan groeit het aantal toestanden explosief. Immers, bij n vriendinnen bestaat een toestand uit een $n \times n$ matrix gevuld met enen en nullen. Weliswaar weten we dat op de diagonaal altijd enen staan, maar dan nog zijn er 2^{n^2-n} mogelijke matrices. Niet al deze matrices zijn bereikbaar (zo geldt bijvoorbeeld altijd dat als i de roddel weet van j er iemand is, naast i zelf, die de roddel weet van i). Toch is er sprake van een enorme *toestandsexplosie*, een exponentiële groei van het aantal bereikbare toestanden als functie van n.

Toen Clarke, Emerson en Sifakis begonnen met hun werk aan model checkers meenden veel collega's dat deze lijn van onderzoek gedoemd was te mislukken: immers bij realistische toepassingen is het aantal toestanden veelste groot om met brute kracht door te rekenen. Met brute rekenkracht, zo meende men, konden systemen met hooguit een miljard toestanden doorgerekend worden, terwijl je voor realistische systemen toch al snel 10^{1000} toestanden hebt. Met grote vasthoudendheid en een enorm doorzettingsvermogen hebben Clarke, Emerson en Sifakis gewerkt aan technieken waarmee toch gerekend kon worden aan praktische toepassingen. Een basisidee is hierbij was om gebruik te maken van zogenaamde *symbolische berekeningen*. Daarbij reken je over een heleboel toestanden tegelijk, net zoals je met de berekening $x^2-y^2 = (x-y)\times(x+y)$ in één keer laat zien dat $36-16 = 2\times 10$, $49-4 = 5\times 9$, enzovoorts. Door symbolisch te rekenen en toestandsverzamelingen compact te representeren met slimme datastructuren, kunnen we immense toestandsruimtes efficiënt doorzoeken.

Zo heeft Emerson veel gewerkt aan het benutten van symmetrie. In Figuur 3 zien we dat de drie toestanden die volgen op de begintoestand eigenlijk allemaal gelijk zijn: twee vriendinnen hebben een roddel uitgewisseld en de derde weet alleen nog haar eigen roddel. Ook de zes toestanden die daarop volgen zijn in essentie allemaal symmetrisch: twee vriendinnen weten alles en de derde weet één roddel nog niet. Door te rekenen met verzamelingen van symmetrische toestanden tegelijk kunnen toestandsexplosies in veel gevallen omzeild worden. Zo hoeft de model checker Uppaal bij 6 vriendinnen slechts 3439 toestanden te doorzoeken met gebruik van symmetriereductie, terwijl het er 839860 zijn zonder gebruik van deze optie.

Naast symmetriereductie hebben onderzoekers nog een talloze technieken bedacht om toestandsexplosies tegen te gaan. Uiteindelijk is het door de combinatie van rekentechnieken dat model checking zich ontwikkeld heeft tot een buitengewoon effectieve technologie waarmee op een efficiënte manier fouten gevonden kunnen worden in complexe ontwerpen.

Toepassingen

Inmiddels zijn er honderden zoniet duizenden aansprekende toepassingen van model checking. Ik noem een paar voorbeelden:

- De meest prominente toepassing van model checkers is zonder twijfel die door Intel op het gebied van hardwareverificatie. In 1994 verloor de chipgigant 475 miljoen dollar door een fout met de drijvendekommadeling in de Pentium-processor. Sindsdien is het gebruik van model checking en andere wiskundige verificatietechnieken uitgegroeid tot een standaard praktijk in de hardware-industrie. Zo is bijvoorbeeld 20 procent van het Pentium IV ontwerp met deze technieken gecontroleerd.

- De SPIN model checker is gebruikt om de "multi-threaded plan execition module" voor de NASA DEEP SPACE 1 missie correct te bewijzen. Hiermee zijn 5 fouten ontdekt die nog niet eerder gevonden waren.

- De SPIN model checker is ook gebruikt om de correctheid te bewijzen van de besturingssoftware van de stormvloedkering bij Rotterdam.

- Mijn student Matthijs Mekking heeft model checking gebruikt om een aantal diepe fouten op te sporen in SHIM6, een nieuwe internetstandaard voor multihoming waarbij als één internetverbinding uitvalt een andere het overneemt zonder dat de gebruiker er iets van merkt.

- Samen met collega's heb ik een aantal fouten gevonden in het Zeroconf protocol, een Internet standaard die gebruikers in staat stelt om locaal een internet netwerk te configureren zonder handmatig IP adressen toe te kennen of gebruik te maken van een DNS server.

- Model checking wordt tegenwoordig wel veel gebruikt bij het analyseren van security protocollen. Bij de recente kraak van de Mifare RFID chip door de Nijmeegse digital security groep is gebruik gemaakt van een fout in het protocol, het geheel van afspraken dat beschrijft hoe de Mifare chip communiceert met zijn omgeving. Indien Philips/NXP indertijd bij het ontwerp van Mifare gebruik zou hebben gemaakt van model checking technieken, dan zou men het huidige lek wellicht al ontdekt hebben voor het in productie nemen van de chip.

Ondanks al deze successen staat het onderzoek op het gebied van model checking nog in de kinderschoenen. Te vaak gebeurt het dat modellen niet doorgerekend kunnen worden vanwege toestandsexplosies. Ook omdat de complexiteit van computersystemen nog steeds groeit, is nog heel veel onderzoek nodig om de effectiviteit van model checkers verder te vergroten.