

Logische Complexiteit

Partiële en Totale Automaten

Jeroen Goudsmit

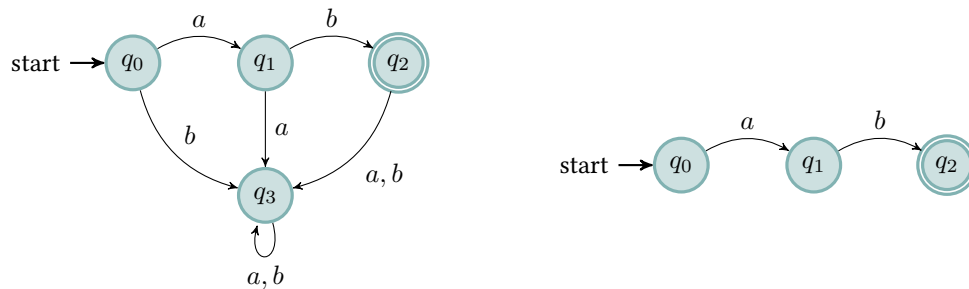
dinsdag 7 februari 2012

In het boek is een (deterministische) eindige automaat (DFA, naar de Engelse naam deterministic finite automaton) gedefinieerd als een tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ waar Q de verzameling der toestanden, Σ het alfabet, q de initiële toestand, δ de transitiefunctie en $F \subseteq Q$ de verzameling der eindtoestanden is. Hier is δ dus een functie

$$\delta : Q \times \Sigma \rightarrow Q.$$

Een functie is doorgaans *totaal* is, wat wil zeggen dat ze gedefinieerd is voor alles in haar domein. Hier betekent dat dus dat $\delta(q, \sigma)$ een toestand aanwijst voor alle $q \in Q$ en $\sigma \in \Sigma$. In een tekening van een DFA kun je dit terugzien aan dat uit iedere toestand er voor ieder symbool in het alfabet precies een uitgaande pijl is.

Dit is soms vervelend. Vaak wil je alleen aangeven waar een woord aan moet voldoen om geaccepteerd te worden, en ben je niet geïnteresseerd in wanneer een woord niet meer van de goede vorm is. Beschouw bijvoorbeeld de automaat hier links onder. Je kan gemakkelijk nagaan dat ze alleen het woord ab accepteren zal. Maar toch moet je omschrijven wat er gebeurt met alle andere woorden. Er naast staat een automaat die alleen omschrijft hoe naar een accepterende toestand gelopen kan worden. In het onderstaande omschrijven we partiële DFA's, die dit idee formeel maken.



Een totale functie $f : X \rightarrow Y$ is zó dat $f(x)$ bestaat voor alle $x \in X$. Een *partiële functie* is een functie waar dit niet voor hoeft te gelden. Dit kunnen we formaliseren door te zeggen dat een partiële functie $f : X \rightarrow Y$ gedefinieerd is als een totale functie $X' \rightarrow Y$ voor een deelverzameling X' van X . Dit betekent dus dat $f(x)$ gedefinieerd is voor alle $x \in X'$, maar omdat X' niet altijd gelijk hoeft te zijn aan heel X , kan $f(x)$ ongedefinieerd zijn, namelijk als $x \in X - X'$.

Definitie 1 (Partiële Deterministische Eindige Automaat). Een partiële deterministische eindige automaat (PDFA naar de Engelse naam, partial deterministic finite automaton) is een tuple $M := \langle Q, \Sigma, \delta, q_0, F \rangle$. Hier zijn Q en Σ eindige verzamelingen, en is F een deelverzameling van Q . De transitiefunctie δ is een partiële functie $Q \times \Sigma \rightarrow Q$.

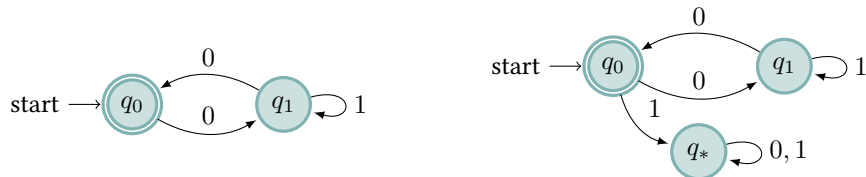
Een woord $w \in \Sigma^*$ wordt *geaccepteerd* precies wanneer er toestanden $q_1, \dots, q_n \in Q$ zijn zodanig dat n de lengte van $w = w_1 w_2 \dots w_n$ is, $q_n \in F$ en met $\delta(q_i, w_{i+1})$ gedefinieerd en gelijk aan q_{i+1} voor alle $0 \leq i < n$. De taal van M , geschreven als $\mathcal{L}(M)$, is de verzameling van woorden die door M geaccepteerd worden.

Uit de bovenstaande definitie is op te maken dat een gebruikelijke (totale) reguliere automaat ook partieel is. De taal die deze automaat accepteert als partiële automaat is precies dezelfde die ze zou accepteren als totale automaat. Voor iedere partiële automaat is er een totale automaat die precies dezelfde taal accepteert.

Lemma 1. *Voor iedere PDFA bestaat er een DFA met dezelfde taal.*

De truc komt neer op het volgende. We maken een nieuwe toestand die dient als afvoerput (in het Engels: *sink*). Alle “ontbrekende transities” worden zo ingevuld als transities naar deze afvoer-toestand. Eenmaal in de afvoer-toestand, kun je er ook niet meer uit ontsnappen. Dit reflecteert dat een woord alleen geaccepteerd wordt indien de automaat tijdens het verwerken van het woord in het “gedefinieerde deel” blijft.

Voorbeeld 1. Beschouw de eindige automaat als hieronder aan de linkerkant, over het alfabet $\Sigma = \{0, 1\}$. Deze automaat is partieel, omdat in toestand q_0 er geen pijl is voor het teken 1. De automaat ernaast is de totale variant, met afvoer-toestand q_* .



Bewijs. Zij $M := \langle Q, \Sigma, \delta, q_0, F \rangle$ een PDFA. Neem een nieuwe toestand q_* (de “afvoerput”) en definieer $\bar{Q}$ als $Q \cup \{q_*\}$. We kunnen nu een nieuwe, totale transitiefunctie $\bar{\delta}$ definiëren op onderstaande wijze.

$$\bar{\delta} : \bar{Q} \times \Sigma \rightarrow \bar{Q}, \quad (q, \sigma) \mapsto \begin{cases} \delta(q, \sigma) & \text{als } \delta(q, \sigma) \text{ gedefinieerd is} \\ q_* & \text{anders} \end{cases}$$

Met deze twee ingrediënten hebben we genoeg om een DFA te maken, namelijk $N := \langle \bar{Q}, \Sigma, \bar{\delta}, q_0, F \rangle$. We claimen nu dat de taal van N gelijk is aan die van M .

Om te bewijzen dat $\mathcal{L}(N) = \mathcal{L}(M)$ nemen we $w_1 w_2 \dots w_n = w \in \Sigma^*$ willekeurig. Stel eerst dat $w \in \mathcal{L}(M)$. Dan betekent dit dat er een rij van toestanden $q_0, \dots, q_n$ is zodanig dat $\delta(q_i, w_{i+1}) = q_{i+1}$ voor alle $0 \leq i < n$, en $q_n \in F$ (een *run* van M). Nu is het ook zo dat $\bar{\delta}(q_i, w_{i+1}) = \delta(q_i, w_{i+1}) = q_{i+1}$, omdat q_i nooit gelijk is aan q_* . Dit betekent dat de automaat N ook het woord w accepteert.

Omgekeerd, als $w \in \mathcal{L}(N)$ dan hebben we ook zo’n pad van toestanden $q_0, \dots, q_n$ waar $q_n \in F$ en $\bar{\delta}(q_i, w_{i+1}) = q_{i+1}$. Merk op dat $q_n \in F$, en omdat $q_* \notin F$ moet het dus wegens de definitie van $\bar{\delta}$ zo zijn dat $q_{n-1} \neq q_*$. Met inductie langs n zien we nu dat $q_0, \dots, q_n \in Q$, en dit toont dat voor alle $0 \leq i < n$ de uitkomst van $\bar{\delta}(q_i, w_{i+1})$ gedefinieerd is en gelijk is aan q_{i+1} , wat op zijn beurt weer hetzelfde is als $\delta(q_i, w_{i+1})$. Zodoende weten we dus ook dat $w \in \mathcal{L}(M)$. $\square$

Opmerking 1. Binnenkort behandelen we NFA’s. Kijk, zodra je de definitie van NFA gezien hebt, nogmaals terug naar dit document en stel jezelf de vraag: “is een PDFA een speciaal soort NFA?”