

Automaten en Berekenbaarheid

Pieter Vanderschueren

Academiejaar 2023-2024

Introductie

Inhoudsopgave

1	Talen en automaten	3
1.1	Wat is een taal?	3
1.2	Een algebra van talen	3
1.3	Reguliere expressies en reguliere talen	4
1.4	Eindige toestandsautomaten	4
1.5	De algebra van NFA's	5
1.6	Van RE naar NFA	6
1.7	Van NFA naar RE	7
1.8	Deterministische eindige toestandsmachines	8
2	Talen en berekenbaarheid	9
3	Herschrijfsystemen	10
4	Andere rekenparadigmas	11
5	Talen en complexiteit	12

1 Talen en automaten

1.1 Wat is een taal?

Definitie 1.1: String over een alfabet Σ

Een **string** over een alfabet Σ is een eindige opeenvolging van nul, één of meer elementen van Σ .

Definitie 1.2: Taal L over een alfabet Σ

Een **taal** L over een alfabet Σ is een verzameling van strings over Σ .

1.2 Een algebra van talen

Definitie 1.3: Een algebra- of algebraïsche structuur

Een algebra- of algebraïsche structuur is een verzameling met daarop een aantal inwendige operaties: dikwijls binaire operaties, maar unair of met grotere ariteit kan ook. Zo wordt de verzameling van alle talen over een alfabet Σ een algebra als we als operaties unie, doorsnede, complement, etc. definiëren. Meer concreet: als L_1 en L_2 twee talen zijn, dan is

- de unie ervan een taal: $L_1 \cup L_2$
- de doorsnede ervan een taal: $L_1 \cap L_2$
- het complement ervan een taal: $\overline{L_1}$

Eigenschap 1.1: Concatenatie van twee talen

Gegeven twee talen L_1 en L_2 over hetzelfde alfabet Σ , dan noteren we de concatenatie van L_1 en L_2 als $L_1 L_2$ en definiëren we:

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

Eigenschap 1.2: De Kleene ster van een taal

De Kleene ster van een taal wordt gedefinieerd als volgt:

$$L^* = \cup_{n \geq 0} L^n$$

1.3 Reguliere expressies en reguliere talen

Definitie 1.4: Reguliere Expressie (RE) over een alfabet Σ

E is een **reguliere expressie** over een alfabet Σ indien E van de vorm is

- ϵ
- ϕ
- a waarbij $a \in \Sigma$
- $(E_1 E_2)$ waarbij E_1 en E_2 reguliere expressies zijn over Σ
- (E_1^*) waarbij E_1 een reguliere expressie is over Σ
- $(E_1 | E_2)$ waarbij E_1 en E_2 reguliere expressies zijn over Σ

Definitie 1.5: Reguliere taal

Een reguliere expressie E bepaalt een **reguliere taal** L_E over hetzelfde alfabet Σ als volgt:

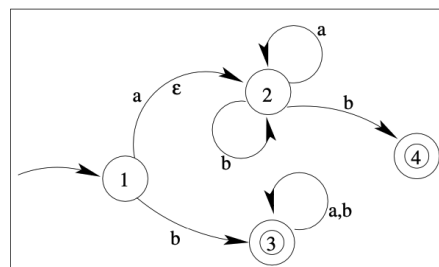
- als $E = a$ (met $a \in \Sigma$) dan is $L_E = \{a\}$
- als $E = \epsilon$ dan is $L_E = \{\epsilon\}$
- als $E = \phi$ dan is $L_E = \emptyset$
- als $E = (E_1 E_2)$ dan $L_E = L_{E_1} L_{E_2}$
- als $E = (E_1)^*$ dan $L_E = L_{E_1}^*$
- als $E = (E_1 | E_2)$ dan $L_E = L_{E_1} \cup L_{E_2}$

1.4 Eindige toestandsautomaten

Definitie 1.6: Niet-deterministische eindige toestandsautomaat (NFA)

Een **niet-deterministische eindige toestandsautomaat** is een 5-tal $(Q, \Sigma, \delta, q_s, F)$ waarbij

- Q een eindige verzameling toestanden is
- Σ is een eindig alfabet
- δ is de overgangsrelatie van de automaat
- q_s is de starttoestand
- $F \subset Q$ is de verzameling eindtoestanden



Definitie 1.7: Een string s wordt aanvaard door een NFA

Een string s wordt aanvaard door een NFA $(Q, \Sigma, \delta, q_s, F)$ indien er een sequentie $q_s = q_0 \xrightarrow{a_0} \dots \xrightarrow{a_{n-1}} q_n$ van overgangen bestaat met $q_n \in F$ zodat s de ϵ -compressiew, wat bekomen wordt door in ϵ te schrappen in de string, is van $a_0 \dots a_{n-1}$.

Dus: Voor toestanden p, q en string $w \in \Sigma^*$ schrijven we $p \xrightarrow{w} q$ indien er een sequentie van overangen $p \xrightarrow{a_0} \dots \xrightarrow{a_{n-1}} q$ bestaat zodat w de ϵ -comprssie is van $a_0 \dots a_{n-1}$.

Definitie 1.8: De taal door een NFA M bepaald

Een taal L wordt bepaald door een NFA M , indien L de verzameling van strings is die M aanvaardt. We noteren de taal van M als L_M .

Definitie 1.9: Equivalentie van twee NFA's

Twee NFA's worden **equivalent** genoemd als ze dezelfde taal bepalen.

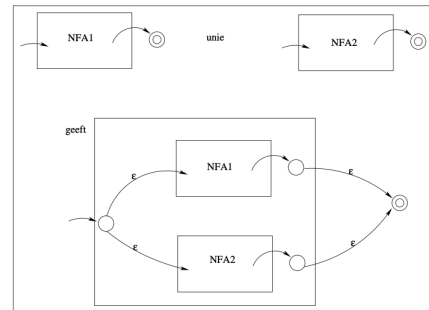
1.5 De algebra van NFA's

Eigenschap 1.3: De unie van twee NFA's

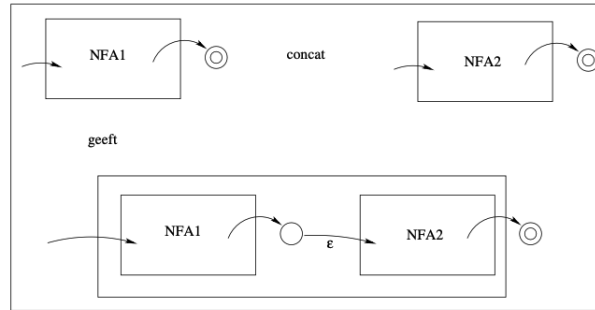
Gegeven: $NFA_1 = (Q_1, \Sigma, \delta_1, q_{s_1}, \{q_{f_1}\})$ en $NFA_2 = (Q_2, \Sigma, \delta_2, q_{s_2}, \{q_{f_2}\})$

De unie $NFA_1 \cup NFA_2$ is de $NFA = (Q, \Sigma, \delta, q_s, F)$ waarbij

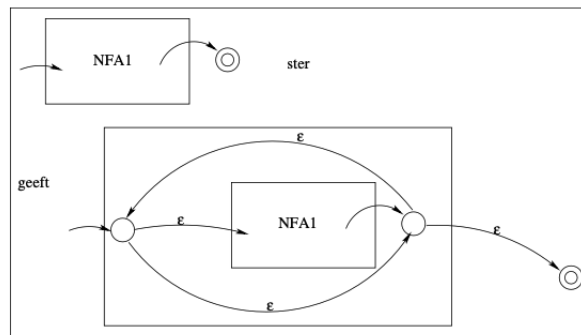
- $Q = Q_1 \cup Q_2 \cup \{q_s, q_f\}$
- $F = \{q_f\}$
- δ is gedefinieerd als:
 - $\forall q \in Q_i \setminus \{q_{f_i}\}, x \in \Sigma, i = 1, 2 : \delta(q, x) = \delta_i(q, x)$
 - $\delta(q_s, \epsilon) = \{q_{s_1}, q_{s_2}\}$
 - $\forall x \in \Sigma : \delta(q_s, x) = \emptyset$
 - $i = 1, 2 : \delta(q_{f_i}, \epsilon) = \{q_f\}$
 - $\forall x \in \Sigma, i = 1, 2 : \delta(q_{f_i}, x) = \emptyset$



Eigenschap 1.4: De concatenatie van twee NFA's



Eigenschap 1.5: De ster van een NFA



1.6 Van RE naar NFA

Definitie 1.10: Van RE naar NFA

We hebben alle ingrediënten om van een reguliere expressie RE een NFA te maken, en zodanig dat de $L_{RE} = L_{NFA}$. Vermits reguliere expressies inductief gedefinieerd zijn zullen we voor elk lijntje van die definitie een overeenkomstige NFA definiëren. De drie basisgevallen, namelijk ϵ , ϕ en $a \in \Sigma$, zijn triviaal te modeleren als NFA. De drie recursieve gevallen beschrijven we als volgt: laat E_1 en E_2 twee reguliere expressies zijn, dan is

- $NFA_{E_1 E_2} = \text{concat}(NFA_{E_1}, NFA_{E_2})$
- $NFA_{E_1^*} = \text{ster}(NFA_{E_1})$
- $NFA_{E_1 | E_2} = \text{unie}(NFA_{E_1}, NFA_{E_2})$

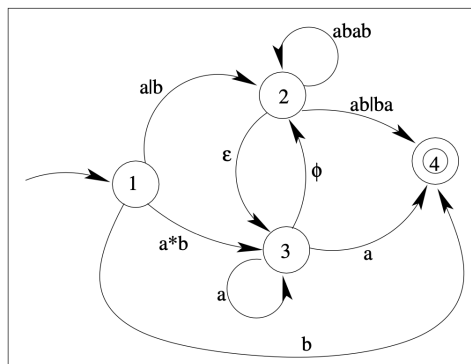
De constructie hierboven bewaart de taal, t.t.z. $L_{NFA_E} = L_E$.

1.7 Van NFA naar RE

Definitie 1.11: GNFA

Een **GNFA** is een eindige toestandsmachine met de volgende wijzigingen en beperkingen:

- er is slechts één eindtoestand en die is verschillend van de starttoestand
- vanuit de starttoestand vertrekt er juist één boog naar elke andere toestand; er komen geen bogen aan in de starttoestand
- in de eindtoestand komt juist één boog aan vanuit elke andere toestand; uit de eindtoestand vertrekken geen bogen
- voor paar p, q (let op: $p = q$ is geldig) van andere toestanden (geen start- of eindtoestand) is er juist één boog $p \rightarrow q$ en één boog $q \rightarrow p$.
- de bogen hebben als label een reguliere expressie



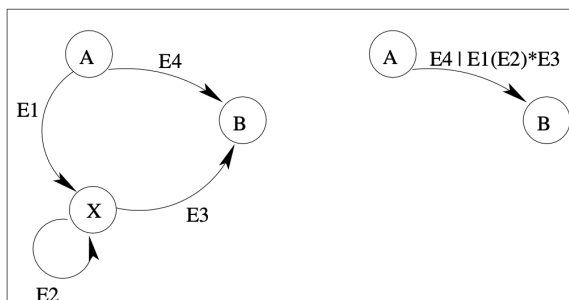
Algoritme 1.1: NFA \rightarrow RE

1. Maak van de NFA een GNFA

- Behoud alle toestanden en bogen van de NFA
- Als er meerdere bogen zijn tussen twee toestanden gelabeld met symbolen $a_1 \dots a_n$ vervang deze door één boog met als label $a_1 | \dots | a_n$
- Voer een nieuwe starttoestand in en een ϵ -boog naar de oude starttoestand
- Voer een nieuwe eindtoestand in en ϵ -bogen vanuit elke oude eindtoestand
- Voor elke boog die ontbreekt tussen twee toestanden om een GNFA te bekomen, voer een ϕ -boog in

2. Reduceer de GNFA:

Kies een willekeurige toestand X verschillend van de start- of eindtoestand, ga naar stap 3 als dit niet mogelijk is. Voor elk paar toestanden A en B (let op: $A = B$ is geldig) verschillend van X bevat de GNFA een unieke boog $A \rightarrow B$ met label E_4 , $A \rightarrow X$ met label E_1 , $X \rightarrow X$ met label E_2 en $X \rightarrow B$ met label E_3 . Vervang het label op de boog $A \rightarrow B$ door $E_4 | E_1 E_2^* E_3$. Doe dit voor alle keuzes voor A en B . Verwijder daarna de knoop X en herhaal.



3. Bepaal RE: de boog van de GNFA heeft als label de gezochte RE

1.8 Deterministische eindige toestandsmachines

Definitie 1.12: Deterministische eindige toestandsmachines

Een NFA is een DFA indien δ geen ϵ -overgangen bevat en indien voor elke $p \in Q$ en elke $a \in \Sigma$ een unieke $q \in Q$ bestaat zodat $p \xrightarrow{a} q$. Het komt erop neer dat in een DFA, δ een totale functie $Q \times \Sigma \rightarrow Q$ is. Voor DFA's zullen we de unieke toestand q zodat $p \xrightarrow{a} q$ dan ook noteren als $\delta(p, a)$.

2 Talen en berekenbaarheid

3 Herschrijfsystemen

4 Andere rekenparadigmas

5 Talen en complexiteit