Automaten en Berekenbaarheid

Pieter Vanderschueren

Academiejaar 2023-2024

Introductie

Inhoudsopgave

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. definïeren. 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_1L_2 en definïeren we:

$$L_1L_2 = \{xy \mid x \in L_1,\ y \in L_2\}$$

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 expressies 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:

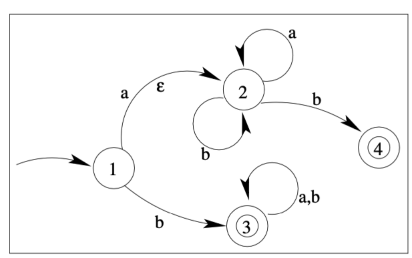
- als $E = a$ (met $a \in \Sigma$) dan is $L_E = \{a\}$
- als $E = \epsilon$ dan is $L_E = \{\epsilon\}$
- als $E = \phi$ dan is $L_E = \emptyset$
- als $E = (E_1 E_2)$ dan $L_E = L_{E_1} L_{E_2}$
- als $E = (E_1)^*$ dan $L_E = L_{E_1}^*$
- als $E = (E_1|E_2)$ dan $L_E = L_{E_1} \cup L_{E_2}$

1.4 Eindge 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 $q_s = q_0 \overset{a_0}{\to} \dots \overset{a_{n-1}}{\to} q_n$ van overgangen bestaat met $q_n \in F$ zodat s de ϵ-compressie, 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 $p \overset{w}{\rightsquigarrow} q$ indien er een sequentie van overangen $p \overset{a_0}{\to} \dots \overset{a_{n-1}}{\to} q$ 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

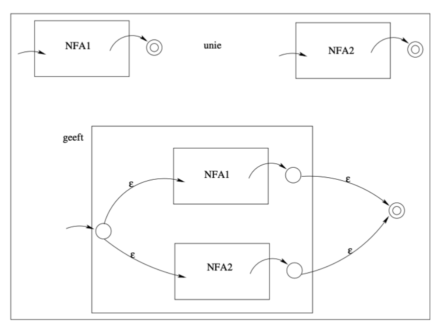
Twee NFA's worden **equivalent** genoemd als ze dezelfde taal bepalen.

1.5 De algebra van NFA's

Eigenschap 1.3: De unie van twee NFA's

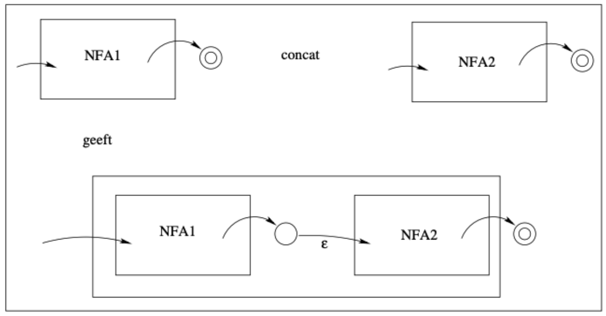
Gegeven: $NFA_1 = (Q_1, \Sigma, \delta_1, q_{s_1}, \{q_{f_1}\})$ en $NFA_2 = (Q_2, \Sigma, \delta_2, q_{s_2}, \{q_{f_2}\})$

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

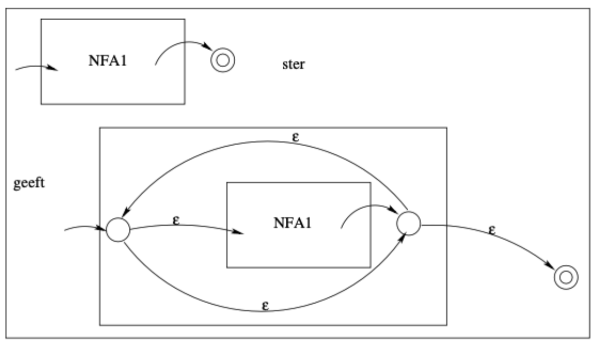


- $Q = Q_1 \cup Q_2 \cup \{q_s, q_f\}$
- $F = \{q_f\}$
- δ is gedefnieerd als:
  - $\forall q \in Q_i \backslash \{q_{f_i}\},\ x \in \Sigma_\epsilon,\ i = 1, 2:\ \delta(q, x) = \delta_i(q, x)$
  - $\delta(q_s, \epsilon) = \{q_{s_1}, q_{s_2}\}$
  - $\forall x \in \Sigma : \delta(q_s, x) = \emptyset$
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

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

- $NFA_{E_1E_2} = \text{concat}(NFA_{E_1}, NFA_{E_2})$
- $NFA_{E_1^*} = \text{ster}(NFA_{E_1})$
- $NFA_{E_1|E_2} = \text{unie}(NFA_{E_1}, NFA_{E_2})$

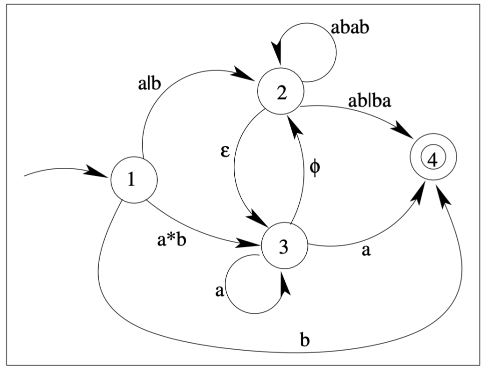
De constructie hierboven bewaart de taal, t.t.z. $L_{NFA_E} = L_E$.

1.7 Van NFA naar RE

Definitie 1.11: GNFA

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

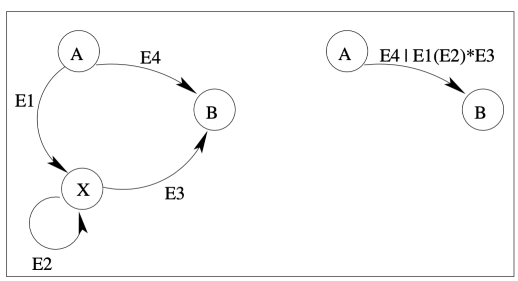
- er is slechts één eindtoestand en die is verschillend van de starttoestand
- vanuit de starttoestand vertrekt er juist één boog naar elke andere toestand; er komen geen bogen aan in de starttoestand
- in de eindtoestand komt juist één boog aan vanuit elke andere toestand; uit de eindtoestand vertrekken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



Algoritme 1.1: NFA → 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 $A \to B$ met label E_4, $A \to X$ met label E_1, $X \to X$ met label E_2 en $X \to B$ met label E_3. Vervang het label op de boog $A \to B$ door $E_4 | E_1 E_2^* E_3$. Doe dit voor alle keuzes voor A en B. Verwijder daana 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 $Q \times \Sigma \to Q$ is. Voor DFA's zullen a we de unieke toestand q zodat $p \xrightarrow{a} q$ dan ook noteren als $\delta(p, a)$.

Stelling 1.1: DFA en NFA equivalentie

Elke NFA is equivalent met een DFA, m.a.w. we kunnen elke NFA $(Q, \Sigma, \delta, q_s, F)$ herleiden tot een equivalente DFA $(Q', \Sigma, \delta', q'_s, F')$ waarbij

- Q': de verzameling van alle deelverzamelingen q' van Q die gesloten zijn onder ϵ-bogen, dus $p \in q' \ \wedge \ p \xrightarrow{\epsilon} q \ \Rightarrow \ q \in q'$
- $\delta' : Q' \times \Sigma \to Q'$
- $q'_s = \{q_s, q \mid q_s \overset{\epsilon}{\rightsquigarrow} q\}$
- $F' = \{q' \in Q' \mid q' \cap F \neq \emptyset\}$

Bewijs 1.1: DFA en NFA equivalentie

Uit constructie volgt dat de geconstrueerde automaat $(Q', \Sigma, \delta', q'_s, F')$ een DFA is. Wat betreft equivalentie, moeten we verifiëren dat $\forall w \in \Sigma^* : q_s \overset{w}{\rightsquigarrow} F \Leftrightarrow q'_s \overset{w}{\rightsquigarrow} F'$. De essentie van dat bewijs is dat voor elke $w \in \Sigma^*$, als in de DFA geldt dat $q'_s \overset{w}{\rightsquigarrow} q$ (in de DFA) dan is $q \ = q_w = \{q \mid q_s \overset{w}{\rightsquigarrow} q\}$ (in de NFA). Dit is eenvoudig inductief te bewijzen gebruik makend van het feit dat $q' = q'_w \Rightarrow \delta'(q', a) = q'_{wa}$. Dan geldt dat de DFA een string w aanvaardt als voor de unieke toestand q' zodat $q'_s \overset{w}{\rightsquigarrow} q'$ geldt dat $q' \cap F \neq \emptyset \ \Leftrightarrow \ q'_w \cap F \neq \emptyset \ \Leftrightarrow$ de NFA aanvaardt w. $\square$

Definitie 1.13: Minimale DFA

Een DFA is minimaal als er geen enkele andere DFA bestaat die dezelfde taal bepaalt en minder toestanden heeft.

Definitie 1.14: f-string

We noemen een string s een f-string vanuit q van de DFA indien $\delta^*(q, s) \in F$, t.t.z. indien er een pad is van q naar een toestand van F die s genereert. F-gelijke toestanden zijn dan toestanden met dezelfde f-strings.

Opmerking: $q \in F \Leftrightarrow \epsilon$ is een f-string vanuit q

Definitie 1.15: f-gelijk

Twee toestanden q_1, q_2 zijn f-gelijk indien

$$\{s \in \Sigma^* \mid \delta^*(q_1, s) \in F\} = \{s \in \Sigma^* \mid \delta^*(q_2, s) \in F\}$$

In woorden, als q_1 en q_2 exact dezelfde f-strings hebben.

Eigenschap 1.6: f-gelijk

- De relatie f-gelijk is een equivalentie-relatie.
- Als p, q f-gelijk zijn dan geldt voor elk symbool a dat $\delta(p, a)$ en $\delta(q, a)$ ook f-gelijk zijn.
- Als p, q f-gelijk zijn dan geldt $p \in F \Leftrightarrow q \in F$.

Bewijs 1.2: Eigenschappen van de f-gelijk relatie

- Het is triviaal om te bewijzen dat f-gelijkheid een equivalentie-relatie is. Dit kan je doen door de reflexiviteit, symmetrie en transitiviteit van de relatie na te gaan.
- Veronderstel dat p, q f-gelijk zijn en veronderstel voor een willekeurig symbool a dat $\delta(p, a) = p', \delta(q, a) = q'$. De f-strings van p en q zijn gelijk, en dus ook hun f-strings van de vorm as. De f-strings van p' zijn de strings s zodat as een f-string is van p. Hetzelfde geldt voor q'. Bijgevolg hebben p', q' dezelfde f-strings en zijn ze dus f-gelijk.
- Als p en q f-gelijk zijn, en $p \in F$ dan is ϵ een f-string van p en dus ook van q. Aangezien er in een DFA geen ϵ-bogen zijn, is $q \in F$. Hetzelfde geldt in de andere richting.

$\square$

Definitie 1.16: Minimale DFA

Een DFA is minimaal als er geen enkele andere DFA bestaat die dezelfde taal bepaalt en minder toestanden heeft, m.a.w. DFA_{min} is een DFA, equivalent met DFA, en alle toestanden zijn f-verschillend.

Stelling 1.2: DFA_{min}

Als een DFA $\text{N} = (Q_1, \Sigma, \delta_1, q_s, F_1)$ een DFA is zonder onbereikbare toestanden en waarin elke twee toestanden f-verschillend zijn, dan bestaat er geen machine met strikt minder toestanden die dezelfde taal bepaalt.

Bewijs 1.3: DFA_{min}

Veronderstel dat $Q_1 = \{q_s, q_1, \ldots, q_n\}$ waarbij q_s de starttoestand is. Stel dat $N_2 = (Q_2, \Sigma, \delta_2, q_s, F_2)$ een DFA is met minder toestanden dan N. Vermits in N elke toestand bereikbaar is, bestaan er strings $\forall i \in \mathbb{N}_0^+ : s_i$ zodanig dat $\delta_1^*(q_s, s_i) = q_i$. Vermits N_2 minder toestanden heeft moet voor een $i \neq j : \delta_2^*(p_s, s_i) = \delta_2^*(p_s, s_j)$. Vermits q_i en q_j f-verschillend zijn, is er een string s zodat $\delta_1^*(q_i, s) \in F_1$ en $\delta_1^*(q_j, s) \notin F_1$ of omgekeerd. Dus ook $\delta_1^*(q_s, s_i s) \in F_1$ en $\delta_1^*(q_s, s_j s) \notin F_1$ of omgekeerd. Dit betekent dat DFA_1 van de strings $s_i s$ en $s_j s$ er juist één accepteert. Maar N_2 zal beide strings $s_i s$ en $s_j s$ accepteren of geen van beiden, aangezien het parsen van s_i en s_j naar dezelfde node leidt, waarna hetzelfde pad gevolgd wordt om v te parsen. Dus kunnen de DFA's N en N_2 niet dezelfde taal bepalen.

$\square$

Definitie 1.17: DFA isomorfisme

Een $\text{DFA}_1 = (Q_1, \Sigma, \delta_1, q_{s_1}, F_1)$ is **isomorf** met een $\text{DFA}_2 = (Q_2, \Sigma, \delta_2, q_{s_2}, F_2)$ als er een bijectie $b : Q_1 \to Q_2$ bestaat zodanig dat

- $b(F_1) = F_2$
- $b(q_{s_1}) = q_{s_2}$
- $b(\delta_1(q, a)) = \delta_2(b(q), a)$

Twee isomorfe DFA's bepalen dus dezelfde taal.

1.9 Myhill-Nerode relaties op Σ^*

2 Talen en berekenbaarheid

3 Herschrijfsystemen

4 Andere rekenparadigmas

5 Talen en complexiteit