AXIOMA

Tentamenbundel

Talen en Automaten

Prijs: f2,00

TENTAMEN

Talen en Automaten

5 maart 1996; 09.00 – 12.00

Je mag gebruik maken van alle stellingen uit het diktaat en van resultaten van opgaven die we deze cursus op het werkcollege hebben behandeld. Geef steeds duidelijk aan wat je gebruikt!

Opgave 1

- Geef de definitie van een contextgevoelige grammatica.
- Geef de definitie van de interpretatie-functie ϕ op de verzameling van reguliere expressies over het alfabet $\{a, b\}$.
- Geef de definitie van een recursieve taal.

Opgave 2

We zeggen dat een taal L over een alfabet T voldoet aan de *PompLemma Eigenschap* (PLE) als geldt:

$$
\begin{array}{l}
\exists(n : n \geq 1 : \\
\quad \forall(z : z \in L \ \wedge \ |z| \geq n : \\
\quad\quad \exists(u, v, w \in T^* :: uvw = z \ \wedge \ |uv| \leq n \ \wedge \ v \neq \varepsilon \ \wedge \\
\quad\quad\quad \forall(t : t \geq 0 : uv^t w \in L))))
\end{array}
$$

Laat nu $L = \{a^i b^j c^k \mid i = 0 \ \vee \ j = k\}$.

- Toon aan: L voldoet aan de PLE.
- Toon aan: L^R voldoet niet aan PLE.
- Toon aan: er zijn niet-reguliere talen die voldoen PLE.

lees verder

Opgave 3

Gegeven de grammatica $G = (\{S\}, \{a, b\}, P, S)$ met

$$P = \{S \to aSbS,\ S \to bSaS,\ S \to \varepsilon\}$$

- Toon aan: G is ambigu.
- Ontwerp een grammatica in Chomsky Normaalvorm voor de taal $\mathcal{L}(G) \setminus \{\varepsilon\}$.
- Welke taal wordt door G gegenereerd? Bewijs je bewering.

Opgave 4

Gegeven zijn de volgende twee beslissingsproblemen:

> **Totaliteitsprobleem voor contextvrije grammatica's (TCFG)**
> **Parameter:** een CFG $G = (N, T, P, S)$.
> **Gevraagd:** $\mathcal{L}(G) = T^*$?

> **Equivalentieprobleem voor contextvrije grammatica's (EqCFG)**
> **Parameter:** twee CFG's G_0 en G_1.
> **Gevraagd:** $\mathcal{L}(G_0) = \mathcal{L}(G_1)$?

Je mag gebruik maken van de kennis dat (**TCFG**) onbeslisbaar is.
Toon aan dat (**EqCFG**) onbeslisbaar is.

Opgave 5

Bewijs:
als $\quad P_0 \in \textsf{NPC} \wedge P_1 \in \textsf{NP} \wedge P_0 \propto P_1 \quad$ dan $\quad P_1 \in \textsf{NPC}$

einde

TENTAMEN

Talen en Automaten

30 augustus 1996; 09.00 – 12.00 uur

Je mag gebruik maken van alle stellingen uit het diktaat en van resultaten van opgaven die we deze cursus op het werkcollege hebben behandeld. Geef steeds duidelijk aan wat je gebruikt!

Opgave 1

- Laat $G = (N, T, P, S)$ een grammatica zijn. Geef de definitie van $\Rightarrow_G$.
- Geef de definitie van een contextvrije grammatica (CFG).
- Geef de definitie van de klasse **NP** .

uitwerking:
Zie dictaat

Opgave 2

Onder het symmetrisch verschil $V \bowtie W$ van twee verzamelingen V en W verstaan we de verzameling

$$V \bowtie W = \{x \mid (x \in V \ \wedge \ x \notin W) \ \vee \ (x \notin V \ \wedge \ x \in W)\}$$

- Bewijs: Als L_0 en L_1 reguliere talen zijn, dan is ook $L_0 \bowtie L_1$ een reguliere taal.

Laat $G = (\{S, A, B\}, \ \ \{a, b\}, \ \ P, \ \ S)$ een reguliere grammatica zijn met P gegeven door de volgende productieregels:

$$\begin{array}{lcl} S & \rightarrow & aA \mid bB \\ A & \rightarrow & bA \mid aB \\ B & \rightarrow & aS \mid \varepsilon \end{array}$$

- Geef een reguliere expressie voor de taal $\mathcal{L}(G)$.

Laat $M = (\{0,1\}, \ \ \{a,b\}, \ \ t, \ \ 0, \ \ \{1\})$ een eindige automaat zijn met t gegeven door

$$t(0,a) = 0, \ \ t(0,b) = 1, \ \ t(1,a) = 1, \ \ t(1,b) = 0$$

- Construeer een DFSA voor de taal $\mathcal{L}(G) \bowtie \mathcal{T}(M)$.

- Laten nu $M_0 = (K_0, T, t_0, b_0, F_0)$ en $M_1 = (K_1, T, t_1, b_1, F_1)$ twee willekeurige DFSA's zijn met totale overgangsfunctie.
  Construeer formeel een DFSA $M_{\bowtie}$ met de eigenschap

$$\mathcal{T}(M_{\bowtie}) = \mathcal{T}(M_0) \bowtie \mathcal{T}(M_1))$$

uitwerking:

•

Merk eerst op

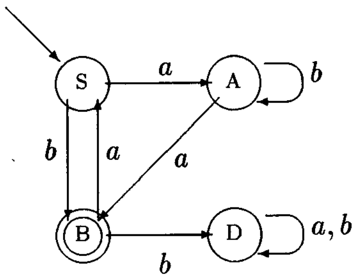
$$V \bowtie W = (V \cap \overline{W}) \cup (\overline{V} \cap W)$$

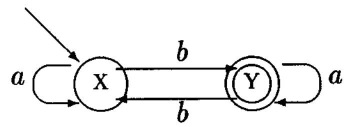
Nu vervolgen we

$$\begin{array}{ll}
 & L_0 \text{ regulier } \ \wedge \ L_1 \text{ regulier} \\
\Rightarrow & \quad \{ \text{ klasse van reguliere talen is gesloten onder complement } \} \\
 & L_0 \text{ regulier } \ \wedge \ \overline{L_1} \text{ regulier } \ \wedge \ \overline{L_0} \text{ regulier } \ \wedge \ L_1 \text{ regulier} \\
\Rightarrow & \quad \{ \text{ klasse van reguliere talen is gesloten onder doorsnede } \} \\
 & (L_0 \cap \overline{L_1} \text{ regulier }) \wedge (\overline{L_0} \cap L_1 \text{ regulier }) \\
\Rightarrow & \quad \{ \text{ klasse van reguliere talen is gesloten onder vereniging } \} \\
 & (L_0 \cap \overline{L_1} \text{ regulier }) \cup (\overline{L_0} \cap L_1 \text{ regulier }) \\
\Rightarrow & \quad \{ \text{ bovenstaande opmerking } \} \\
 & L_0 \bowtie L_1 \text{ regulier}
\end{array}$$

•
Uit onderstaande automaat voor $\mathcal{L}(G)$ halen we de reguliere expressie

$$(ab^*aa + ba)^*(ab^*a + b)$$





•
We construeren een transitietabel voor een automaat voor de taal $\mathcal{L}(G) \bowtie \mathcal{T}(M)$ vanuit de twee automaten die hierboven zijn afgedrukt. Het idee is om een automaat te maken die simultaan de twee bovenstaande automaten simuleert. Is de hele invoerstring afgewerkt (en daarom is de eerste automaat ook totaal gemaakt) dan kunnen we kijken in welke toestand elk van de twee automaten zich bevindt. We moeten de string accepteren indien precies één van de twee automaten in een accepterende toestand is uitgekomen.

	a	b
SX	AX	BY
AX	BX	AY
BY	SY	DX
BX	SX	DY
AY	BY	AX
SY	AY	BX
DX	DX	DY
DY	DY	DX

De gevraagde automaat is nu

$$(\{SX, SY, AX, AY, BX, BY, DX, DY\}, a, b, t, SX, \{SY, AY, DY, BX\})$$

met t gegeven door de vorige tabel.

•
Gebaseerd op het concrete voorbeeld uit het vorige onderdeel komen we op de automaat

$$(K_0 \times K_1, T, t, (b_0, b_1), (F_0 \times (K_1 \setminus F_1)) \cup ((K_0 \setminus F_0) \times F_1))$$

met t gegeven door: voor elke $k_0 \in K_0$, $k_1 \in K_1$ en $a \in T$:

$$t((k_0, k_1), a) = (t_0(k_0, a), t_1(k_1, a))$$

[redacted] Opgave 3

Gegeven de grammatica $G = (\{S\},\ \{a,b\},\ P,\ S)$ met

$$P = \{S \rightarrow aSbS,\ S \rightarrow bSaS,\ S \rightarrow \varepsilon\}$$

- Toon aan: G is ambigu.
- Ontwerp een grammatica in Chomsky Normaalvorm voor de taal $\mathcal{L}(G) \setminus \{\varepsilon\}$.
- Welke taal wordt door G gegenereerd? Bewijs je bewering.

uitwerking:

•

We geven twee afleidingsbomen voor de string *abab*

```
          S                                S
   ------|------                    ------|------
   |    |    |    |                 |    |    |    |
   a    S    b    S                 a    S    b    S
        |      ---|---                ---|---      |
              a S b S                a S b S
                |   |                  |   |
```

Hieruit blijkt dat G ambigu is.

•

Voordat we het algoritme uit kunnen voeren zullen we eerst de ε-productie moeten verwerken:

$S \rightarrow aSbS,\ \ S \rightarrow abS,\ \ S \rightarrow aSb,\ \ S \rightarrow ab,\ \ S \rightarrow bSaS,\ \ S \rightarrow baS,\ \ S \rightarrow bSa,\ \ S \rightarrow ba$

Dan nu naar de Chomsky normaalvorm: Primaire productie-regels zijn er niet, zodat we gelijk met stap 2 van start kunnen: de secundaire productieregels verwijderen.

$$\begin{array}{lcl} S & \rightarrow & ASBS \mid ABS \mid ASB \mid AB \mid BSAS \mid BAS \mid BSA \mid BA \\ A & \rightarrow & a \\ B & \rightarrow & b \end{array}$$

En tot slot de tertiare productieregels:

$$\begin{array}{lcl} S & \rightarrow & AK \mid AL \mid AM \mid AB \mid BX \mid BY \mid BZ \mid BA \\ A & \rightarrow & a \\ B & \rightarrow & b \\ K & \rightarrow & SL \\ L & \rightarrow & BS \\ M & \rightarrow & SB \\ X & \rightarrow & SY \\ Y & \rightarrow & AS \\ Z & \rightarrow & SA \end{array}$$

•
Bewering:
$$\mathcal{L}(G) = \{w \in \{a,b\}^* \mid \#a.w = \#b.w\}$$

We zullen dit bewijzen door wederzijdse inclusie.

$\mathcal{L}(G) \subseteq \{w \in \{a,b\}^* \mid \#a.w = \#b.w\}$

Inductie naar de lengte van de afleiding:

$$P(n) :: \forall(w \in \{a,b\}^* :: S \Rightarrow^n w \;\Rightarrow\; \#a.w = \#b.w)$$

Basis: $n = 1$

$$\begin{array}{ll} & S \Rightarrow^1 w \\ \Rightarrow & \{\text{ eenstaps afleiding komt overeen met productieregel }\} \\ & (S, w) \in P \\ \Rightarrow & \{\text{ inspectie productieregels }\} \\ & w = \varepsilon \\ \Rightarrow & \{\text{ tellen }\} \\ & \#a.w = \#b.w \end{array}$$

Stap: $(n \geq 1)$

$$\begin{array}{ll} & S \Rightarrow^{n+1} w \\ \Rightarrow & \{\; n+1 \geq 2;\text{ splits de eerste stap af }\} \\ & \exists(\alpha \in \{S,a,b\}^* :: S \Rightarrow \alpha \Rightarrow^n w) \\ \Rightarrow & \{\text{ eenstapsafleiding; inspectie productieregels; gevalsonderscheid }\} \\ & \exists(u, v \in \{a,b\}^*, k, m \in \mathbb{N} : k, m < n : (S \Rightarrow aSbS \;\wedge\; S \Rightarrow^k u \;\wedge\; S \Rightarrow^m v \;\wedge\; aubv = w) \\ & \qquad\qquad\qquad \vee\; (S \Rightarrow bSaS \;\wedge\; S \Rightarrow^k u \;\wedge\; S \Rightarrow^m v \;\wedge\; buav = w) \\ \Rightarrow & \{\text{ inductiehypothese }\} \\ & \exists(u, v \in \{a,b\}^*, k, m \in \mathbb{N} : k, m < n : (\#a.u = \#b.u \;\wedge\; \#a.v = \#b.v \;\wedge\; aubv = w) \\ & \qquad\qquad\qquad \vee\; (\#a.u = \#b.u \;\wedge\; \#a.v = \#b.v \;\wedge\; buav = w) \\ \Rightarrow & \{\text{ tellen }\} \\ & \#a.w = \#b.w \end{array}$$

$\{w \in \{a,b\}^* \mid \#a.w = \#b.w\} \subseteq \mathcal{L}(G)$

We bewijzen deze inclusie door inductie naar de lengte van de string:

$$P(n) :: \forall(w \in \{a,b\}^* : \#a.w = \#b.w : |w| = n \;\Rightarrow\; S \Rightarrow^* w)$$

Basis: $(n = 0)$

$$\begin{array}{ll} & |w| = 0 \\ \Rightarrow & \{\;\} \\ & w = \varepsilon \end{array}$$

$\Rightarrow$ { inspectie productieregels }
$(S, w) \in P$

$\Rightarrow$ { eenstapsafleiding volgt uit productieregel }
$S \Rightarrow^1 w$

$\Rightarrow$ { }
$S \Rightarrow^* w$

Stap: $(n > 0)$
Neem aan dat w begint met een a; het andere geval gaat analoog.
Als $w = av$ en $\#a.w = \#b.w$, dan zal v voldoen aan $\#a.v = \#b.v - 1$. Bekijk nu de functie f die bij elke prefix x van v de waarde van $\#b.x - \#a.x$ geeft. Dan is f een geheelwaardige functie met $f(\varepsilon) = 0$ en $f(v) = 1$. Dat betekent dat er een kleinste prefix y van v bestaat met $f(y) = 1$. Maar dat houdt in dat y eindigt op een b: y was de *kleinste* prefix met $f(y) = 1$, dus alle echte prefixen van y hebben een kleinere f-waarde. Met andere woorden: $y = zb$ met $\#a.z = \#b.z$.
We kunnen dus v opdelen tot $v = zbt$ met $\#a.z = \#b.z$ en als gevolg daarvan $\#a.t = \#b.t$. Van deze eigenschap zullen we hieronder gebruik maken.

$|w| = n$

$\Rightarrow$ { opmerking hierboven }
$\exists(z, t \in \{a, b\}^* :: (w = azbt \ \vee \ w = bzat) \wedge \#a.z = \#b.z \wedge \#a.t = \#b.t)$

$\Rightarrow$ { inductiehypothese }
$\exists(z, t \in \{a, b\}^* :: (w = azbt \ \vee \ w = bzat) \wedge S \Rightarrow^* z \wedge S \Rightarrow^* t)$

$\Rightarrow$ { distributie; inspectie productieregels }
$\exists(z, t \in \{a, b\}^* :: (S \Rightarrow aSbS \Rightarrow^* azbS \Rightarrow^* azbt = w)$
$\vee (S \Rightarrow bSaS \Rightarrow^* bzaS \Rightarrow^* bzat = w))$

$\Rightarrow$ { eliminatie }
$S \Rightarrow^* w$

Afrondend:

$w \in \mathcal{L}(G)$

$\equiv$ { definitie taal }
$w \in \{a, b\}^* \wedge S \Rightarrow^* w$

$\equiv$ { bovenstaande inductiebewijzen }
$w \in \{a, b\}^* \wedge \#a.w = \#b.w$

$\equiv$ { verz.leer }
$w \in \{x \in \{a, b\}^* \mid \#a.x = \#b.x\}$

Opgave 4

Gegeven het volgende beslissingsprobleem:

> **Inverseprobleem voor Turingmachines (InvTM)**
> **Parameter:** een Turingmachine M met invoeralfabet T en een string $y \in T^*$.
> **Gevraagd:** Is er een $x \in T^*$ met $\mathcal{F}_M(x) = y$?

Is (**InvTM**) beslisbaar? Bewijs je bewering.

uitwerking:

(**InvTM**) is onbeslisbaar.
We bewijzen dit door een reductie te maken van het Haltingprobleem naar het Inversieprobleem.
Laat (M, w) een instantie zijn van (**HP**). Construeer de twee-band TM M' als volgt:

> schrijf w op band 2;
> simuleer M op band 2;
> geef w op band 1 als uitvoer;

Dan is $R((M, w)) = (M', w)$ een instantie van (**InvTM**).
Merk nu op:

$$
\begin{array}{ll}
 & (M, w) \in \mathcal{Y}_{(\mathsf{HP})} \\
\Rightarrow & \{ \text{ def. } (\mathsf{HP}) \text{ } \} \\
 & M \text{ stopt op } w \\
\Rightarrow & \{ \text{ simulatie eindigt } \} \\
 & \mathcal{F}_{M'}(w) = w \\
\Rightarrow & \{ \text{ eenpuntsregel } \} \\
 & \exists (x \in T^* :: \mathcal{F}_{M'}(x) = w) \\
\Rightarrow & \{ \text{ def. } (\mathsf{InvTM}) \text{ } \} \\
 & (M', w) \in \mathcal{Y}_{(\mathsf{InvTM})}
\end{array}
$$

$$
\begin{array}{ll}
 & (M, w) \in \mathcal{N}_{(\mathsf{HP})} \\
\Rightarrow & \{ \text{ definitie } (\mathsf{HP}) \text{ } \} \\
 & M \text{ stopt niet op } w \\
\Rightarrow & \{ \text{ simulatie eindigt niet } \} \\
 & \forall (x \in T^* :: \mathcal{F}_{M'}(x) \text{ ongedefinieerd }) \\
\Rightarrow & \{ \text{ } \} \\
 & \neg\exists (x \in T^* :: \mathcal{F}_{M'}(x) = w) \\
\Rightarrow & \{ \text{ definitie } (\mathsf{InvTM}) \text{ } \} \\
 & (M', w) \in \mathcal{N}_{(\mathsf{InvTM})}
\end{array}
$$

Dus R is inderdaad een reductie van (**HP**) naar (**InvTM**). Op grond van de reductiestelling en de onbeslisbaarheid van het Haltingprobleem is nu de onbeslisbaarheid van het Inversieprobleem bewezen.

Opgave 5

Toon aan:

$$\text{als} \quad \mathsf{P} \cap \mathsf{NPC} \neq \emptyset \quad \text{dan} \quad \mathsf{P} = \mathsf{NP}$$

uitwerking:

Uit de definities van P en NP volgt direct dat $\mathsf{P} \subseteq \mathsf{NP}$.

We bewijzen nu de omgekeerde inclusie. Neem aan dat $\mathsf{P} \cap \mathsf{NPC} \neq \emptyset$.
Laat dan $X \in \mathsf{P} \cap \mathsf{NPC}$.
Laat nu $Y \in \mathsf{NP}$.
Uit $X \in \mathsf{NPC}$ volgt dat er een reductie van polynomiale tijdcomplexiteit R bestaat van Y naar X.
Uit $X \in \mathsf{P}$ volgt dat er een algoritme van polynomiale tijdcomplexiteit A bestaat voor X.
De samenstelling van A en R is nu een algoritme van polynomiale tijdcomplexiteit voor Y.
Samenvattend: voor elke $Y \in \mathsf{NP}$ is er algoritme van polynomiale tijdcomplexiteit te construeren, dus $\mathsf{NP} \subseteq \mathsf{P}$.

Deze twee inclusies samen bewijzen de gelijkheid van de twee verzamelingen.

einde