Karakterisation af oversættelse for de til Chomsky hierarkiet hørende automater

Erik Meineche Schmidt

DAIMI PB - 3
Marts 1972

DATALOGISK AFDELING
AARHUS UNIVERSITET
Ny Munkegade, Bygn. 540
8000 Aarhus C, Denmark

Karakterisation af oversættelse for de til Chomsky-hierarkiet hørende automater

Erik Meineche Schmidt

1972

INDLEDNING

Nærværende skrift behandler følgende problemstilling:

> Givet en af de til Chomsky-hierarkiet hørende automater, f.eks. en push-down acceptor. Forsyn denne med output, dvs. sørg for at hver gang automaten 'foretager' sig noget, giver det anledning til at der på en envejs output tape skrives en streng over et alfabet. Definer grafen for en sådan maskine som mængden af par (x, y), hvor input x dels tilhører det sprog, automaten accepterer, dels giver anledning til output y. Hvorledes kan sådanne grafer karakteriseres for de forskellige typer af automater?

Historisk set starter behandlingen af den emnekreds, der er beslægtet med ovenstående, med Ginsburg og Rose [1966], som indfører begrebet transducere, dvs. automater uden sluttilstande, der anvendes som en slags generaliserede afbildninger på sprog.

Lewis og Stearns [1968] beskæftiger sig med oversættelse af såvel deterministiske som ikke-deterministiske context-free sprog, Aho og Ullman [1969a] indfører begrebet syntax-dirigeret oversættelsesskema for context-free sprog, medens Ibarra [1971] er den første til at anvende de af Ginsburg og Greibach [1969] introducerede Familier af Abstrakte Sprog og Automater. I en helt ny artikel behandler Culik II og Morey [1971] bl.a. for første gang oversættelse defineret af lineært begrænsede automater.

Den her givne fremstilling skal tjene to formål. Dels som speciale i forbindelse med hovedfagseksamen i datalogi og dels som materiale i forbindelse med et 2. dels kursus i Formelle Sprog og Automater. Der er derfor her og der medtaget beviser, som er "overflødige" i den forstand, at de i forvejen findes i litteraturen.

INDHOLD

AFSNIT 1

NOTATION OG LIGNENDE

Vi vil i vid udstrækning anvende samme notation som Hopcroft og Ullman [1969].

Således vil vi – med mindre andet udtrykkeligt er angivet – altid have at

- store bogstaver betegner hjælpesymboler
- små bogstaver i begyndelsen af alfabetet betegner terminale symboler
- små bogstaver i slutningen af alfabetet betegner strenge af terminale symboler
- små græske bogstaver betegner strenge bestående af såvel hjælpe- som terminale symboler.

Et alfabet er altid en endelig ikke tom mængde af symboler. Hvis Σ er et alfabet, er Σ^* som sædvanlig mængden af strenge over Σ. Hvis vi betegner det tomme ord med ϵ er $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$.

Hvis D er en mængde, betyder $|D|$ antallet af elementer i D og $\mathcal{P}_{fin}(D)$ betyder mængden af endelige delmængder af D.

Hvis α er en streng, er $|\alpha|$ antallet af elementer i strengen.

Lad Σ, Δ være to alfabeter. En homomorfi h er en afbildning $h : \Sigma^* \to \Delta^*$ som opfylder

$$h(\epsilon) = \epsilon$$
$$\forall x, y \in \Sigma^* : h(xy) = h(x)\, h(y)$$

En grammatik er et system $G = (V_N, V_T, P, S)$ hvor V_N er hjælpesymboler, V_T er terminale symboler, med $V = V_N \cup V_T$ er P produktioner af formen $\alpha \to \beta$, $\alpha \in V^+$, $\beta \in V^*$, og endelig er $S \in V_N$ startsymbolet. Relationen $\Rightarrow$ mellem strenge i V^* defineres ved at $\alpha \Rightarrow \beta$, hvis der findes $\mu, \delta \in V^*$ således at $\alpha = \mu\, \omega\, \delta$, $\beta = \mu\, \eta\, \delta$ og $\omega \rightarrow \eta$ er en produktion. Betegner vi med $\overset{*}{\Rightarrow}$ den transitive, refleksive lukning af $\Rightarrow$, er sproget genereret af G mængden $L(G) = \{ w \in V_T^* \mid S \overset{*}{\Rightarrow} w \}$.

En grammatik som den ovenfor definerede er altid en Turing grammatik.

Lad G være en Turing grammatik. Såfremt der for enhver produktion $\alpha \rightarrow \beta$ gælder at $|\beta| \geq |\alpha|$ er G context sensitiv.

Lad G være en Turing grammatik. Såfremt der for enhver produktion $\alpha \rightarrow \beta$ gælder at $\alpha \in V_N$ er G context-free.

Lad G være en context-free grammatik. Såfremt det for enhver produktion $A \rightarrow \beta$ gælder at $\beta \in (V_T \cup \{\varepsilon\}) \cdot (V_N \cup \{\varepsilon\})$, er G regulær.

Bemærk, at det iflg. ovenstående er umuligt for et context sensitivt sprog at indeholde det tomme ord. Dette kommer man normalt ud over ved at tillade produktionen $S \rightarrow \varepsilon$ forudsat at S ikke optræder på højresiden af nogen produktion. Af tekniske årsager vil vi imidlertid her og i det følgende fastholde den oprindelige definition.

Klasserne af sprog hørende til de fire typer af grammatikker vil vi betegne med hhv. RE, CS, CF og REG.

Da vi senere får brug for Post Correspondence Problem (PCP), vil vi indføre dette her:

Lad Σ være et alfabet med $|\Sigma| \geq 2$ og lad $A = \{x_1, \ldots, x_k\}$, $B = \{w_1, \ldots, w_k\}$ være lister bestående af k strenge $x_i, w_i \in \Sigma^+$, $1 \leq i \leq k$. Sådanne lister vil vi kalde PCP-lister.

Vi siger, at PCP med lister A og B har en løsning, såfremt der findes heltal $i_1, \dots, i_m$, $m \geq 1$ med $i_j \leq k$, $1 \leq j \leq m$ således at $x_{i_1} \dots x_{i_m} = w_{i_1} \dots w_{i_m}$. I så fald skriver vi PCP(A, B) = 1 og ellers PCP(A, B) = 0. Det er velkendt, at PCP er rekursivt uløseligt.

Lad os slutte dette afsnit med at anføre, at vi overalt i fremstillingen vil bruge symbolet □ til at afslutte sætninger, lemmaer og corrollarer samt beviser herfor.

AFSNIT 2

ABSTRAKT FAMILIE AF SPROG OG ABSTRAKT FAMILIE AF AUTOMATER

Vi har følgende definition fra Ginsburg og Greibach [1969]:

Definition 2. 1 En Abstrakt Familie af sprog (AFL) er et par $(\Sigma, \mathcal{L})$ eller $\mathcal{L}$ hvis Σ er underforstået hvor

i) Σ er en uendelig mængde af symboler

ii) $\forall L \in \mathcal{L} \ \exists \Sigma_1 \subseteq \Sigma \ \ |\Sigma_1| < \infty : L \subseteq \Sigma_1^*$

iii) $\mathcal{L}$ er lukket under operationerne $\cup$, $\cdot$, $+$, ϵ-free homomorfi, invers homomorfi og snit med regulære mængder

iv) $\exists L \in \mathcal{L} : L \neq \emptyset$ □

Definition 2. 2 En AFL $\mathcal{L}$ siges at være full, såfremt $\mathcal{L}$ er lukket under vilkårlig homomorfi. □

Følgende lemma er bevist overalt i litteraturen. Vi viser det alligevel her, fordi vi senere vil gøre kraftigt brug af det.

Lemma 2. 3 Lad $L \in RE$. Der findes da et $L_1 \in CS$ og en homomorfi h således at $h(L_1) = L$.

Bevis Lad $G = (V_N, V_T, P, S)$ være en grammatik med $L(G) = L$ og lad # være et symbol, som ikke er med i $V_N \cup V_T$. Vi kan uden indskrænkning antage, at alle produktioner i P er på formen

$$\begin{array}{ll} \alpha \to \beta & \alpha, \beta \in V_N^* \\ A \to a & A \in V_N, \ a \in V_T \end{array}$$

Betragt grammatikken $G_1 = (V_N, V_T \cup \{\#\}, P_1, S)$, som har følgende produktioner

i) $\alpha \to \beta \in P_1$ hvis $\alpha \to \beta \in P \wedge |\beta| \geq |\alpha|$

ii) $\alpha \to \beta \#^k \in P_1$ hvis $\begin{cases} \alpha \to \beta \in P \wedge |\beta| < |\alpha| \\ k = |\alpha| - |\beta| \end{cases}$

iii) $\# A \to A\#$ for alle $A \in V_N$

Definer h ved

$$h(a) = \begin{cases} a & \text{hvis } a \in V_T \\ \epsilon & - \quad a = \# \end{cases}$$

Det er klart, at $h(L_1) = L$ □

Betragtes Chomsky-hierarkiet, er det velkendt, at hver af klasserne REG, CF, RE er en full AFL, medens CS kun er en AFL.

Næste skridt er at knytte en familie af automater til en familie af sprog. Vi har følgende definition fra Ginsburg og Greibach [1969].

<u>Definition 2.4</u> En Abstrakt Familie af (envejs non-deterministiske) Acceptorer (AFA) er et par $(\Omega, \mathfrak{D})$ eller $\mathfrak{D}$ hvis Ω er underforstået med følgende egenskaber

i) Ω er et system $(K, \Sigma, \Gamma, I, f, g)$ hvor

a) K, Σ er uendelige mængder og Γ, I er mængder med $\Gamma \neq \emptyset$, $I \neq \emptyset$

b) f er en afbildning $f : \Gamma^* \times I \to \Gamma^* \cup \{\emptyset\}$

c) g er en afbildning $g : \Gamma^* \to \mathcal{P}_{fin}(\Gamma^*)$ således at $g(\epsilon) = \epsilon$ og $\epsilon \in g(\gamma)$ hvis og kun hvis $\gamma = \epsilon$

d) for ethvert $\gamma \in g(\Gamma^*)$ findes der $1_\gamma \in I$ så $f(\gamma', 1_\gamma) = \gamma'$ for alle γ' med $\gamma \in g(\gamma')$

e) for ethvert $u \in I$ findes der en endelig mængde $\Gamma_u \subseteq \Gamma$ således at hvis $\Gamma_1 \subseteq \Gamma$, $\gamma \in \Gamma_1^*$ og $f(\gamma, u) \neq \emptyset$ så er $f(\gamma, u) \in (\Gamma_1 \cup \Gamma_u)^*$

ii) $\mathfrak{D}$ er familien af elementer (acceptorer) af formen
$D = (K_1, \Sigma_1, \delta, q_o, F)$ hvor

a) K_1, Σ_1 er endelige delmængder af K, Σ. $F \subseteq K_1$, $q_o \in K_1$

b) δ er en afbildning
$\delta : K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times g(\Gamma^*) \to \mathcal{P}_{fin}(K_1 \times I)$
således at mængden
$G_D = \{\gamma \mid \exists\, q \in K_1, \exists\, a \in \Sigma_1 \cup \{\epsilon\} : \delta(q, a, \gamma) \neq \emptyset\}$
er endelig. □

<u>Definition 2.5</u> Lad $(\Omega, \mathfrak{D})$ være en AFA og lad $D = (K_1, \Sigma_1, \delta, q_o, F)$ være en acceptor. $\vdash$ er en relation på $K_1 \times \Sigma_1^* \times \Gamma^*$ defineret ved:

For $a \in \Sigma_1 \cup \{\epsilon\}$ er $(q, ax, \gamma) \vdash (p, x, \gamma')$ såfremt der findes $\bar{\gamma}$ og u med $\bar{\gamma} \in g(\gamma)$, $(p, u) \in \delta(q, a, \bar{\gamma})$ og $f(\gamma, u) = \gamma'$.

Lad $\vdash^*$ være den reflexive, transitive lukning af $\vdash$. Ved sproget accepteret af D vil vi forstå mængden

$$L(D) = \{w \in \Sigma_1^* \mid \exists\, p \in F : (q_o, w, \epsilon) \vdash^* (p, \epsilon, \epsilon)\}$$

Vi definerer også $\mathcal{L}(\mathfrak{D}) = \{L(D) \mid D \in \mathfrak{D}\}$ □

Eksempler på hvorledes nogle af de sædvanlige typer af automater falder inden for denne definition, findes i Ginsburg og Greibach [1969]. Det er også her, vi finder hovedsætningen om sammenhængen mellem AFL og AFA, nemlig

<u>Sætning 2.6</u> Lad $\mathcal{F}$ være en AFL. Der findes da en AFA $\mathfrak{D}$ således at $\mathcal{F} = \mathcal{L}(\mathfrak{D})$ hvis og kun hvis $\mathcal{F}$ er en full AFL. □

Heraf følger umiddelbart, at der til familierne REG, CF, RE findes AFA'er med de i sætning 2.6 nævnte egenskaber. Disse AFA'er er familierne af endelige automater, push down automater og Turing maskiner.

AFSNIT 3

ABSTRAKT FAMILIE AF OVERSÆTTERE

En oversætter (translator) er i denne sammenhæng en acceptor, der er udstyret med en output-tape og et dertil hørende output-alfabet. Hver gang maskinen "foretager sig noget", giver den som output et ord over output-alfabetet.

Eksempel 3.1 Lad $M = (K, \Sigma, \Gamma, \delta, q_o, Z_o, F)$ være en push-down automat som defineret i Hopcroft og Ullmann [1969]. Transitionsfunktionen δ er en afbildning

$$\delta : K \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathcal{P}_{fin}(K \times \Gamma^*)$$

En push-down translator $M' = (K, \Sigma, \Gamma, \Delta, \delta', q_o, Z_o, F)$ fås ved at tilføje output-alfabetet Δ og udvide billedmængden for δ til

$$\delta' : K \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathcal{P}_{fin}(K \times \Gamma^* \times \Delta^*)$$ □

Dette generaliseres i følgende definitioner fra Ibarra [1971].

Definition 3.2 En abstrakt familie af (envejs non-deterministiske) oversættere (AFTR) er et par $(\Omega, \mathcal{M})$ eller $\mathcal{M}$ hvis Ω er underforstået hvor

i) Ω er et system $(K, \Sigma, \Gamma, I, f, g)$ med samme egenskaber som i def. 2.4.

ii) $\mathcal{M}$ er familien af elementer (oversættere) $M = (K_1, \Sigma_1, \Sigma_2, \delta, q_o, F)$ hvor

a) K_1, Σ_1, q_o, F er som i def. 2.4

b) $\Sigma_2 \subseteq \Sigma$ med $0 < |\Sigma_2| < \infty$

c) δ er en afbildning
$\delta : K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times g(\Gamma^*) \to \mathcal{P}_{fin}(K_1 \times I \times \Sigma_2^*)$
således at mængden
$G_M : \{\gamma \mid \exists a \in \Sigma_1 \cup \{\epsilon\}, \exists q \in K_1 : \delta(q, a, \gamma) \neq \emptyset\}$
er endelig. □

Definition 3.3 Lad $\mathcal{M}$ være en AFTR og lad $M = (K_1, \Sigma_1, \Sigma_2, \delta, q_o, F)$ være en oversætter i $\mathcal{M}$. $\vdash$ er en relation på $K_1 \times \Sigma_1^* \times \Gamma^* \times \Sigma_2^*$ defineret ved

for $a \in \Sigma_1 \cup \{\varepsilon\}$ er $(q, ax, \gamma, y) \vdash (p, x, \gamma', yv)$ såfremt der findes $\bar{\gamma}$ og u med $\bar{\gamma} \in g(\gamma)$ $(p, u, v) \in \delta(q, a, \bar{\gamma})$ og $f(\gamma, u) = \gamma'$

Lad $\overset{*}{\vdash}$ være den reflexive, transitive lukning af $\vdash$. Ved grafen for M vil vi forstå mængden

$$Gr(M) = \{(x,y) \mid \exists p \in F : (q_o, x, \varepsilon, \varepsilon) \overset{*}{\vdash} (p, \varepsilon, \varepsilon, y)\}$$

og vi sætter $Gr(\mathcal{M}) = \{Gr(M) \mid M \in \mathcal{M}\}$ □

Definition 3.4 Lad M være en oversætter. Ved definitionsmængden for M vil vi forstå mængden

$$Dom(M) = \{x \mid (x,y) \in Gr(M)\}$$

og ved billedmængden

$$Ran(M) = \{y \mid (x,y) \in Gr(M)\}$$ □

Definition 3.5 Lad $(\Omega, \mathcal{M})$ være en AFTR og lad $(\Omega_1, \mathfrak{A})$ være en AFA. $\mathcal{M}$ og $\mathfrak{A}$ siges at være associerede såfremt $\Omega = \Omega_1$. □

Hvis $\mathcal{M}$ er en AFTR, har vi følgende sætning fra Ibarra [1971], der karakteriserer $Gr(\mathcal{M})$.

Sætning 3.6 Lad $\mathcal{M}$ være en AFTR og lad $\mathfrak{A}$ være dens associerede AFA. Lad Σ_2, Σ_3 være endelige delmængder af Σ, og lad $T \subseteq \Sigma_2^* \times \Sigma_3^*$. Der gælder da, at der findes $M \in \mathcal{M}$, således at $T = Gr(M)$ hvis og kun hvis der findes en endelig mængde $\Sigma_1 \subseteq \Sigma$, et sprog $L \in \mathcal{L}(\mathfrak{A})$, $L \subseteq \Sigma_1^*$ og to homomorfier $h_1 : \Sigma_1^* \to \Sigma_2^*$, $h_2 : \Sigma_1^* \to \Sigma_3^*$ således at

$$T = \{(h_1(w), h_2(w)) \mid w \in L\}.$$

Bevis

a) Lad $T \subseteq \Sigma_2^* \times \Sigma_3^*$ og lad $M_1 = \{K_1, \Sigma_2, \Sigma_3, \delta, q_o, F\}$ være i $\mathcal{M}$ med $T = Gr(M_1)$.

Sæt $n = \max\{|y| \mid (p, u, y) \in \delta_1(q, a, \gamma), (q, a, \gamma) \in K_1 \times (\Sigma_2 \cup \{\varepsilon\}) \times G_{M_1}\}$. At n eksisterer, følger af def. 3.2 ii) – c). Sæt $\Sigma' = \{[y] \mid |y| \leq n\} \subseteq \Sigma \setminus (\Sigma_2 \cup \Sigma_3)$. Sæt $K' = \{[p, y] \mid (p, u, y) \in \delta_1(q, a, \gamma), (q, a, \gamma) \in K_1 \times (\Sigma_2 \cup \{\varepsilon\}) \times G_{M_1}\} \subseteq K \setminus K_1$.

Foreløbig er der blevet knyttet et nyt symbol $[y]$ til hver mulig output streng, og vi har skabt en mængde af tilstande $[p, y]$, som husker hvilket output, der skulle udsendes samtidig med overgang til tilstanden.

Vi definerer nu følgende acceptor
$D_1 = (K_1 \cup K', \Sigma_2 \cup \Sigma', \delta_2, q_o, F)$ ved

i) $\forall (q, a, \gamma) \in K_1 \times (\Sigma_1 \cup \{\varepsilon\}) \times G_{M_1}$:

$$\begin{array}{l} (p, u, y) \in \delta_1(q, a, \gamma) \\ \Updownarrow \\ ([p, y], u) \in \delta_2(q, a, \gamma) \end{array}$$

ii) $\forall ([p, y], [y], \gamma') \in K_3 \times \Sigma' \times G_{M_1}$:

$$(p, 1\gamma') = \delta_2([p, y], [y], \gamma')$$

Denne konstruktion tvinger os til at bruge reglerne fra i) og ii) alternerende. Yderligere er vi nødt til at starte med en regel af type i) og slutte med en af type ii). Heraf følger, at hvis $w \in L(D_1)$ kan w skrives på formen

$$w = a_1[y_1]\ a_2[y_2] \ldots a_k[y_k]$$

hvor $a_i \in \Sigma_1 \cup \{\varepsilon\}$ og $[y_i] \in \Sigma'$.

Af konstruktionen følger det imidlertid også, at så er $(a_1 \ldots a_k, y_1 \ldots y_k) \in Gr(M_1)$. Antag nemlig at vi i D_1 anvender en regel fra i) på a_i. Den eneste måde hvorpå ii) kan anvendes er at vi vi møder $[y]$, hvor y er output fra M_1 under anvendelse af den den tilsvarende regel her.

Lad omvendt $(x, y) \in Gr(M_1)$. Det betyder pr. definition, at der findes en følge af konfigurationer for M_1 $k_1 \dots k_m$ med $k_1 = (q_o, x, \epsilon, \epsilon)$ $k_m = (p, \epsilon, \epsilon, y)$ $p \in F$, således at

$$k_i \vdash k_{i+1} \qquad 1 \le i \le m-1$$

og k_i, k_{i+1} er givet ved $k_i = (q', a_i w, \gamma, u)$ $k_{i+1} = (p', w, \gamma', uv_i)$ hvor $a_i \in (\Sigma_2 \cup \{\epsilon\})$ $u, v_i \in \Sigma_3^*$. Så kan vi skrive x hhv. y på formen $x = a_1 \dots a_m$ $y = v_1 \dots v_m$ men det betyder at $a_1 [v_1] \dots a_m [v_m] \in L(D_1)$ ifølge konstruktionen af D_1.

Hvis H er mængden af elementer (a, y) for hvilke der findes p, q, γ, u, således at $\delta(p, a, \gamma) \ni (q, u, y)$, har vi vist at

$$L(D_1) = \{a_1 [y_1] \dots a_k [y_k] \mid (a_1 \dots a_k, y_1 \dots y_k) \in (T(M_1) \wedge (a_i, y_i) \in H, \quad 1 \le i \le k\}$$

Da nu $\mathcal{M}$ og $\mathfrak{D}$ var associerede, gælder at $D_1 \in \mathfrak{D}$ dvs. $L(D_1) \in \mathcal{L}(\mathfrak{D})$.

Sæt $\Sigma_1 = (\Sigma' \cup \Sigma_2)$ og definer homomorfierne h_1, h_2 ved

$$h_1(c) = \begin{cases} c \text{ hvis } c \in \Sigma_2 \\ \epsilon \text{ ellers} \end{cases}$$

$$h_2(c) = \begin{cases} y \text{ hvis } c = [y] \in \Sigma' \\ \epsilon \text{ ellers} \end{cases}$$

hvorefter vi har at

$$Gr(M) = \{(h_1(w), h_2(w)) \mid w \in L(D_1)\}$$

b) Lad nu $L \in \mathcal{L}(\mathfrak{D})$ $L \subseteq \Sigma_1^*$ og lad $h_1: \Sigma_1^* \to \Sigma_2^*$, $h_2: \Sigma_1^* \to \Sigma_3^*$ være to homomorfier. Definer Σ' ved

$\Sigma' = \{[h_1(a), h_2(a)] \mid a \in \Sigma_1\} \subseteq \Sigma \setminus (\Sigma_1 \cup \Sigma_2 \cup_3)\}$ og definer homomorfien h': $\Sigma_1{}^* \to \Sigma'^*$ ved $h'(a) = [h_1(a), h_2(a)]$.
Da $\mathcal{L}(\mathfrak{D})$ er en AFL, gælder der at $h'(L) \in \mathcal{L}(\mathfrak{D})$.
Lad $D = \{K_1, \Sigma', \delta_1, q_o, F\}$ være en acceptor med $L(D) = h'(L)$.
Lad os definere en oversætter $M = \{K_1, \Sigma', \Sigma_3, \delta_2, q_o, F\}$ ved:

i) $\forall (q, [x,y], \gamma) \in K_1 \times \Sigma' \times G_D$:

$$(p,u) \in \delta_1(q, [x,y], \gamma)$$
$$\Updownarrow$$
$$(p,u,y) \in \delta_2(q, [x,y], \gamma)$$

ii)
$$(p,u) \in \delta_1(q, \epsilon, \gamma)$$
$$\Updownarrow$$
$$(p,u,\epsilon) \in \delta_2(q, \epsilon, \gamma)$$

Da $\mathfrak{D}$ og $\mathcal{M}$ er associerede. gælder der at $M \in \mathcal{M}$, og det følger klart af konstruktionen at

$$Gr(M) = \{([h_1(a_1), h_2(a_1)] \ldots [h_1(a_n), h_2(a_n)], h_2(a_1 \ldots a_n)) \mid a_1 \ldots a_n \in L\}$$

Betragt følgende homomorfi h: $\Sigma'^* \to \Sigma_2^*$ defineret ved $h[h_1(a), h_2(a)] = h_1(a)$. Såfremt vi kan vise, at der findes $M_1 \in \mathcal{M}$ således at

$$Gr(M_1) = \{(h(x), y) \mid (x,y) \in Gr(M)\}$$

er beviset for sætning 3.6 fuldført. Dette vises af det følgende lemma. $\square$

<u>Lemma 3.7</u> Lad $\mathcal{M}$ være en AFTR og lad $M \in \mathcal{M}$ være en oversætter med $Gr(M) \subseteq \Sigma_1{}^* \times \Sigma_2{}^*$. Lad h: $\Sigma_1{}^* \to \Sigma_3{}^*$ være en homomorfi. Der findes da $M_1 \in \mathcal{M}$ således at

$$Gr(M_1) = \{(h(x), y) \mid (x,y) \in Gr(M)\}$$

<u>Bevis</u> Lad $M = \{K_1, \Sigma_1, \Sigma_2, \delta_1, q_o, F\}$ og sæt $\Sigma_1' = \{a \in \Sigma_1 \mid h(a) = \epsilon\}$.

Lad $\Sigma_1 \setminus \Sigma_1' = \{a_1 \ldots a_n\}$ og lad $h(a_i) = b_{i_1} \ldots b_{i_{k_i}}$ $1 \leq i \leq n$.
Lad endelig $K' = \{p_{i_1}, \ldots, p_{i_{k_i}}\}_{i=1}^{n}$ være elementer i $K \setminus K_1$.
Definer oversætteren $M_1 = \{K_1 \cup K_1 \times K', \Sigma_3, \Sigma_2, \delta_1, q_o, F\}$
ved

i) $\delta(q, \epsilon, \gamma) \ni (q', u, y) \Updownarrow \delta_1(q, \epsilon, \gamma) \ni (q', u, y)$

ii) $\delta(q, a, \gamma) \ni (q', u, y)$ for $a \in \Sigma_1'$ $\Updownarrow \delta_1(q, \epsilon, \gamma) \ni (q', u, y)$

iii) $\delta(q, a_i, \gamma) \ni (q', u, y)$ for $a_i \in \Sigma_1 \setminus \Sigma_1'$

$$\Updownarrow \begin{cases} \delta_1\ (q, b_{i_1}, \gamma) \ni ([q', p_{i_1}], 1\gamma, \epsilon) \\ \delta_1\ ([q', p_{i_m}], b_{i_{m+1}}, \gamma) \ni ([q', p_{i_{m+1}}], 1\gamma, \epsilon) \quad 1 \leq m \leq k_i - 1 \\ \delta_1\ ([q', p_{i_{k_i}}], \epsilon, \gamma) \ni (q', u, y) \end{cases}$$

Det er let at indse, at M_1 har de ønskede egenskaber. □

Bemærk, at det først er i beviset for lemma 3.7, at vi anvender at $\mathscr{L}(\mathfrak{D})$ er en <u>full</u> AFL. Dette skal forstås på følgende måde:

Da $\mathcal{M}$ og $\mathfrak{D}$ er associerede, er Dom (M) $\in \mathscr{L}(\mathfrak{D})$. M_1 som konstrueret i lemma 3.7 har Dom(M_1) = h (Dom(M)), og da vi kræver, at $M_1 \in \mathcal{M}$ er det nødvendigt at h(Dom(M)) $\in \mathscr{L}(\mathfrak{D})$ dvs. at $\mathscr{L}(\mathfrak{D})$ er lukket over for vilkårlig homomorfi.

Hvis man forsøger at definere familier af automater hørende til en AFL som ikke er full, viser det sig (Ginsburg og Greibach [1969]), at disse stort set er karakteriseret ved at de kun må foretage et begrænset antal sammenhængende ϵ-træk. Af regel ii) i beviset for lemma 3.7 ses det, at selvom M opfylder dette kriterium, er det ikke sikkert, at M_1 gør det. Det er lige præcis det, der er problemet, når vi i afsnit 4 skal til at se på oversættelser defineret af lineært begrænsede automater.

<u>Definition 3.8</u> En push-down translator (PDTR) er et system M = $\{K, \Sigma, \Gamma, \Delta, \delta, q_o, Z_o, F\}$ hvor

K er en endelig mængde af tilstande
Σ - - - - - inputsymboler
Γ - - - - - staksymboler
Δ - - - - - outputsymboler
$q_o \in$ er starttilstanden
$Z_o \in \Gamma$ er stakbunden
$F \subseteq K$ er mængden af sluttilstande.
δ er en afbildning
δ: $K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_{fin} (K \times \Gamma^* \times \Delta^*)$

$\vdash$ er følgende relation på $K \times \Sigma^* \times \Gamma^* \times \Delta^*$

for $a \in \Sigma \{\varepsilon\}$ er $(q, ax, \gamma Z, u) \vdash (p, x, \gamma\gamma', uv)$
såfremt $(p, \gamma', v) \in \delta(q, a, Z)$.

Hvis $\vdash^*$ er den reflexive, transitive lukning af $\vdash$, definerer vi grafen af M ved

$$Gr(M) = \{(x, y) \mid \exists p \in F : (q_o, x, Z_o, \varepsilon) \vdash^* (p, \varepsilon, \varepsilon, y)\} \qquad \Box$$

Definition 3.9 En generaliseret sekventiel maskine (GSM) er et system $M = (K, \Sigma, \Delta, \delta, q_o, F)$ hvor K, Σ, q_o, F er som i def. 3.8. δ er en afbildning

$$\delta: K \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}_{fin} (K \times \Delta^*)$$

$\vdash$ er følgende relation på $K \times \Sigma^* \times \Delta^*$

for $a \in \Sigma \cup \{\varepsilon\}$ $(q, ax, u) \vdash (p, x, uv)$
såfremt $(p, v) \in \delta(q, a)$

$\vdash^*$ er igen den reflexive transitive lukning af $\vdash$ og vi sætter grafen for M til

$$Gr(M) = \{(x, y) \mid \exists p \in F : (q_o, x, \varepsilon) \vdash^* (p, \varepsilon, y)\} \qquad \Box$$

Ginsburg og Greibach [1969] viser, hvorledes familierne af push-down automater og endelige automater hver for sig danner en AFA. Ved hjælp heraf er det trivielt at vise at familierne af PDTR og GSM'er danner AFTR'er, men det giver straks følgnede corollar til sætning 3.6.

<u>Corollar 3.10</u> Lad Σ, Δ være to alfabeter og lad $T \subseteq \Sigma^* \times \Delta^*$. der findes da en PDTR (GSM) M således, at Gr(M) = T hvis og kun hvis der findes et sprog $L \in CF$ (REG) og to homomorfier h_1, h_2 således at

$$T = \{(h_1(w), h_2(w)) \mid w \in L\} \qquad \square$$

Lad os vende os mod en anden måde at karakterisere grafer for PDTR og GSM'er, nemlig v. hj. a. grammatikker. Aho og Ullman [1969a] indfører begrebet syntaxdirigeret oversættelses skema (SDTS). Her er vi kun interesseret i den simpleste udgave af SDTS, nemlig det såkaldte simpel syntax dirigeret oversættelses skema (SSDTS).

<u>Definition 3.11</u> En SSDTS er et system $G = (V, \Sigma, \Delta, R, S)$, hvor V er en endelig mængde af hjælpesymboler, Σ, Δ, er endelige mængder af hhv. input- og outputsymboler, $S \in V$ er startsymbolet og R er en endelig mængde af regler. Vi har som sædvanlig $V \cap (\Sigma \cup \Delta) = \emptyset$.

En form er et par (α, β) med $\alpha \in (V \cup \Sigma)^*$, $\beta \in (V \cup \Delta)^*$ som opfylder at hvis vi betragter homomorfien $h: (V \cup \Sigma \cup \Delta)^* \to V^*$ defineret ved

$$h(A) = \begin{cases} A & \text{hvis } A \in V \\ \epsilon & \text{ellers} \end{cases}$$

så er $h(\alpha) = h(\beta)$.

En regel er et element af formen $A \to (\alpha, \beta)$ hvor $A \in V$ og (α, β) er en form. $\square$

(α, β) er altså en form hvis og kun hvis α og β indeholder de samme hjælpesymboler i samme rækkefølge.

Definition 3.12 Lad G være en SSDTS. Lad os definere relationen $\Rightarrow$ mellem former for G. For former (α_1, β_1) (α_2, β_2) skriver vi

$$(\alpha_1, \beta_2) \Rightarrow (\alpha_2, \beta_2)$$

såfremt vi kan skrive (α_1, β_1) på formen $(\omega_1 \, A \, \delta_1, \; \omega_2 \, A \, \delta_2)$ med $h(\omega_1) = h(\omega_2)$, $h(\delta_1) = h(\delta_2)$ og vi kan skrive (α_2, β_2) på formen $(\omega_1 \, \gamma_1 \, \delta_1, \; \omega_2 \, \gamma_2 \, \delta_2)$ og der gælder at $A \rightarrow (\gamma_1, \gamma_2)$ er en regel.

Hvis $\overset{*}{\Rightarrow}$ er den reflexive transitive lukning af $\Rightarrow$ vil vi ved den simple syntax dirigerede oversættelse (SSDT) defineret af G forstå mængden

$$T(G) = \{ (x, y) \mid (S, S) \overset{*}{\Rightarrow} (x, y) \} \qquad \square$$

Når vi anvender en regel i en SSDTS erstatter vi altså samtidigt samme forekomst af samme hjælpesymbol i en form med regelens højreside.

Ved hjælp af Corollar 3.10 kan vi nu vise følgende sætning fra Aho og Ullman [1969a].

Sætning 3.13 Lad Σ, Δ være alfabeter og lad $T \subseteq \Sigma^* \times \Delta^*$. Der findes da en PDTR M således at $T = Gr(M)$ hvis og kun hvis der findes en SSDTS $G = (V, \Sigma, \Delta, R, S)$ således at $T = T(G)$.

Bevis Lad M være en PDTR med $Gr(M) = T$. Lad $L \in CF$, h_1, h_2 homomorfier så

$$Gr(M) = \{ (h_1(w), \; h_2(w)) \mid w \in L \}$$

og lad $G_L = (V_N, V_T, P, S)$ være en grammatik for L. Konstruer $G = (V_N, \Sigma, \Delta, R, S)$ på følgende måde

Hvis $A \to x_1 B_1 \ldots x_n B_n x_{n+1}$ $n \geq 0$
er et element i P er

$$A \to (h_1(x_1) B_1 \ldots h_1(x_n) B_n h_1(x_{n+1}), h_2(x_1) B_1 \ldots h_2(x_n) B_n h_2(x_{n+1}))$$

element i R.

Det er klart at

$$T = \{(h_1(w), h_2(w)) \mid w \in L\} = T(G)$$

Lad omvendt $G = (V, \Sigma, \Delta, R, S)$ være en SSDTS. Konstruer en grammatik $G_L = (V, V_T, P, S)$ på følgende måde

hvis $A \to (x_1^i B_1^i \ldots x_{n_i}^i B_{n_i}^i x_{n_i+1}^i, y_1^i B_1^i \ldots y_{n_i}^i B_{n_i}^i y_{n_i+1}^i)$
er den i'te regel i R er

$$A \to c_1^i B_1^i \ldots c_{n_i}^i B_{n_i}^i c_{n_i+1}^i$$

en produktion i P. $\{c_1^i \ldots c_{n+1}^i\}$ er forskellige symboler for hver regel i R. Sæt $C = \bigcup_i \{c_1^i, \ldots c_{n_i+1}^i\}$ og definer to homomorfier $h_1 : C^* \to \Sigma^*$, $h_2 : C^* \to \Delta^*$ ved

$$h_1(c_j^i) = x_j^i \qquad h_2(c_j^i) = y_j^i$$

Lad M være den PDTR, som har

$$Gr(M) = \{(h_1(w), h_2(w)) \mid w \in L(G_L)\}$$

Det er klart at $Gr(M) = T(G)$. □

Hvis vi definerer en analogi til SSDTS for regulære grammatikker ved at en form har udseendet

$$(\alpha, \beta) = (xA, yA) \quad x \in \Sigma^*, y \in \Delta^*, A \in V \cup \{\varepsilon\}$$

og på indlysende vis definerer analogien til SSDT, er det lige ud ad landevejen at vise en sætning for GSM'er, der er analog til sætning 3. 13.

Hermed sluttes omtalen af oversættelsesbegrebet på grundlag af context-free sprog. Aho og Ullman [1969] omhandler almindelig syntax dirigeret oversættelse dvs. oversættelse, hvor komponenterne i en form (α, β) stadig skal indeholde de samme hjælpesymboler, men hvor rækkefølgen i β kan være en permutation af rækkefølgen i α. Aho og Ullman [1971] generaliserer begrebet SDTS ved til hver knude i et derivationstræ for et ord at knytte ikke én men flere semantiske variable.

AFSNIT 4

LINEÆRT BEGRÆNSEDE AUTOMATER OG LINEÆRT BEGRÆNSEDE OVERSÆTTERE.

Der findes rundt om i litteraturen forskellige definitioner af lineært begrænsede automater. Disse er dog alle ækvivalente i den forstand, at klassen af sprog, der accepteres, er CS. Her vil vi bruge næsten samme definition som i Ginsburg og Rose [1966].

Definition 4.1 En lineært begrænset automat (LBA) er et system $M = (K, \Sigma, \Gamma, \delta, q_o, F)$ hvor

K er en endelig ikke tom mængde af tilstande
Σ – – – – – – – inputsymboler
$\Gamma \supseteq \Sigma$ er en endelig ikke tom mængde af tapesymboler
$q_o \in K$ er starttilstanden
$F \subseteq K$ er mængden af sluttilstande
δ er en afbildning

$$\delta : K \times \Gamma \rightarrow \mathcal{P}_{fin}(K \times \Gamma \times \{L, N, R\}) \qquad \square$$

Definition 4.2 Lad $M = (K, \Sigma, \Gamma, \delta, q_o, F)$ være en LBA. $\vdash$ er en relation på $\Gamma^* K \Gamma^*$ defineret ved

for $u, v, \in \Gamma^*$, $c \in \Gamma$ er

i) $ucpav \vdash uqcbv$
$\Updownarrow$
$\delta(p, a) \ni (q, b, L)$

ii) $upav \vdash uqbv$
$\Updownarrow$
$\delta(p, a) \ni (q, b, N)$

iii) $upav \vdash ubqv$
$\Updownarrow$
$\delta(p, a) \ni (q, b, R)$

Hvis $\vdash^*$ er den reflexive, transitive lukning af $\vdash$ vil vi ved sproget accepteret af M forstå mængden

$$T(M) = \{ w \in \Sigma^+ \mid \exists\, q \in F,\ \alpha \in \Gamma^* : q_o\, w \vdash^* \alpha\, q \} \qquad \square$$

Bemærk, at der for alle LBA M gælder, at $\varepsilon \notin T(M)$. (jfr. side 1.2)

Definition 4.3 En lineært begrænset oversætter (LBTR) er et system $M = (K, \Sigma, \Gamma, \Delta, \delta, q_o, F)$ hvor $K, \Sigma, \Gamma, q_o, F$ er som i def. 4.1 og der yderligere gælder

Δ er en endelig ikke tom mængde af output symboler

δ er en afbildning

$$\delta : K \times \Gamma \to \mathcal{P}_{fin}(K \times \Gamma \times (L, N, R) \times \Delta^*) \qquad \square$$

Definition 4.4 Lad $M = (K, \Sigma, \Gamma, \Delta, \delta, q_o, F)$ være en LBTR. $\vdash$ er en relation på $\Gamma^* K \Gamma^* \times \Delta^*$ defineret ved

for $u, v \in \Gamma^*$, $c \in \Gamma$, $x, y \in \Delta^*$ er

i) $(ucpav, x) \vdash (uqcbv, xy)$
$\Updownarrow$
$\delta(p, a) \ni (q, b, L, y)$

ii) $(upav, x) \vdash (uqbv, xy)$
$\Updownarrow$
$\delta(p, a) \ni (q, b, N, y)$

iii) $(upav, x) \vdash (ubqv, xy)$
$\Updownarrow$
$\delta(p, a) \ni (q, b, R, y)$

Hvis $\vdash^*$ igen er den reflexive, transitive lukning af $\vdash$ vil vi ved grafen for M forstå mængden

$$Gr(M) = \{(w, y) \in \Sigma^+ \times \Delta^* \mid \exists\, q \in F,\ \exists\, \beta \in \Gamma^* : (q_o w, \varepsilon) \vdash^* (\beta q, y)\} \qquad \square$$

Bemærkning 4.5 Af ovenstående definitioner følger det, at hvis M er en LBTR, er $\mathrm{Dom}(M) \in CS$. $\square$

Som kommentar til def. 4.1-4.4 kan det siges, at der ingen steder optræder de "begrænsnings mærker", som er velkendte fra andre definitioner af LBAïer. Sådanne steder optræder der to specielle symboler, f.eks. $¢$ og $\$$ som bruges til at afgrænse den del af båndet, hvorpå LBAïen må arbejde. Havde vi taget begrænsningsmærker med i vor definition af en LBA M, skulle vi have lagt følgende krav på δ

i) $\delta(q, ¢) \ni (p, b, X) \Downarrow b = ¢ \wedge X = R$

ii) $\delta(q, \$) \ni (p, b, X) \Downarrow b = \$ \wedge X = L$

og vi ville i så fald have defineret $T(M)$ ved

$$T(M) = \{ w \in \Sigma^+ | \exists q \in F, \exists \beta \in \Gamma^* : ¢ q_o w \$ \vdash^* ¢ \beta q \$ \}$$

Ginsburg og Rose [1966] har imidlertid vist, at man ikke opnår noget ekstra ved at medtage begrænsningsmærker, dvs. den klasse af sprog der accepteres er i begge tilfælde klassen af context sensitive sprog. I det følgende kan vi derfor i flæng bruge den af de to definitioner af LBA og LBTR, der "passer" bedst til lejligheden.

I lys af ovenstående bemærkninger kan vi nu give et simpelt bevis for et resultat (Ginsburg og Rose [1966]), der viser en af forskellene på grafer for LBTRïer og de i afsnit 3 omtalte oversættere.

<u>Sætning 4.6</u> Lad $L \in RE$. $L \subseteq \Sigma^*$. Der findes da en LBTR M, således at $Ran(M) = L$.

<u>Bevis</u> Lad L_1 være som i lemma 2.3 og lad $M_1 = (K, \Sigma \cup \{\#\}, \Gamma, \delta_1, q_o, F)$ være en LBA med begrænsningsmærker, således at $L_1 = T(M_1)$. Definer en LBTR $M = (K \cup \{p_o, p_1, p_2\}, \Sigma \cup \{\#\}, \Gamma \cup \Gamma \times \Sigma, \Sigma, \delta, q_o, p_2)$ ved

for $q, q' \in K$, $a \in \Sigma$, $b, c \in \Gamma$, $X \in \{L, N, R\}$

i) $\delta(q_o, a) = (p_o, (a, a), R, \epsilon)$
ii) $\delta(p_o, a) = (p_o, (a, a), R, \epsilon)$
iii) $\delta(p_o, \$) = (p_o, \$, L, \epsilon)$
iv) $\delta(p_o, (a, a)) = (p_o, (a, a), L, \epsilon)$

v) $\delta(p_0, ¢) = (q_0, ¢, R, \varepsilon)$

vi) $\delta(q, (a,b)) \ni (q', (c,b), x, \varepsilon)$

$\Updownarrow$ $\delta_1(q, b) \ni (q', c, x)$

vii) $\delta(q, \$) = (p_1, \$, L, \varepsilon)$ for $q \in F$

viii) $\delta(p_1, (a,b)) = (p_1, (a,b), L, \varepsilon)$

ix) $\delta(p_1, ¢) = (p_2, ¢, R, \varepsilon)$

x) $\delta(p_2, (a,b)) = (p_2, a, R, b)$ $b \in \Sigma$

$\delta(p_2, (a, \#)) = (p_2, a, R, \varepsilon)$

M starter med at gemme en kopi af input i andenkomponenten af symbolerne (.,.) og derefter simulerer den M_1 på førstekomponenterne. Såfremt M_1 accepterer kopierer M_1 alle oprindelige inputsymboler på nær $\#$. Det er klart at Ran(M) = L . □

I det følgende vil vi se på, i hvor høj grad resultaterne fra afsnit 3 kan overføres til LBTRier. Inden vi er helt klar til det, skal vi imidlertid have endnu en bemærkning om begrænsningsmærker.

Definition 4.7 En context sensitiv grammatik med begrænsningsmærker, er et system $G = (V_N, V_T, P, S, \#)$ hvor

i) $\# \notin V_T$

ii) $(V_N, V_T \cup \{\#\}, P, S)$ er en context sensitiv grammatik

iii) For enhver produktion $\alpha \to \beta$ gælder der at α, β samtidigt er på en af følgende former $\#\delta$, δ, $\delta\#$ hvor $\delta \in V^*$

Ved sproget genereret af G vil vi forstå mængden

$$\# L \# = L(G) = \{ \# w \# \mid \# S \# \overset{*}{\Rightarrow} \# w \# \}$$ □

Landweber [1963] har vist, at L er context sensitiv hvis og kun hvis $\# L \#$ genereres af en context sensitiv grammatik med begrænsningsmærker, således at vi også her kan anvende den form, der passer bedst i det givne tilfælde.

Vi kan nu vise følgende:

Sætning 4. 8 Lad $M = (K, \Sigma, \Gamma, \Delta, \delta, q_o, F)$ være en LBTR og lad Λ være et symbol med $\Lambda \notin (\Sigma \cup \Delta)$. Der findes et context sensitivt sprog $L \subseteq (\Sigma \cup \Delta \cup \{\Lambda\})^*$ og to homomorfier $h_1 : (\Sigma \cup \Delta \cup \Lambda)^* \to \Sigma^*$, $h_2 : (\Sigma \cup \Delta \cup \Lambda)^* \to \Delta^*$ således at

$$Gr(M) = \{ (h_1(w), h_2(w)) \mid w \in L \} \qquad \square$$

Beviset for sætning 4. 8 er temmelig teknisk, så vi viser det v. hj. a. et par lemmaer.

Lemma 4. 9 Lad $M = (K, \Sigma, \Gamma, \delta, q_o, F)$ være en LBA. Konstruer en context sensitiv grammatik $G = (\Gamma \times \Sigma \cup K \times \Gamma \times \Sigma, \Sigma, P, S, \#)$ hvor P er følgende produktioner

i) $S \to S\,(a, a)$, $S \to (a, a)$ for alle $a \in \Sigma$

ii) $\#\,(a, a) \to \#\,[q_o, a, a]$

iii) $(p, b, L) \in \delta(q, a) \Updownarrow (d, e)\,[q, a, c] \to [p, d, e]\,(b, c)$

iv) $(p, b, N) \in \delta(q, a) \Updownarrow [q, a, c] \to [p, b, c]$

v) $(p, b, R) \in \delta(q, a)$ for $p \in F$ $\Updownarrow [q, a, c]\,(d, e) \to (b, c)\,[p, d, e]$

vi) $(p, b, R) \in \delta(q, a)$ for $p \in F$ $\Updownarrow [q, a, c]\,\# \to c\,\#$

vii) $(d, e)\,c \to e\,c$

og hvor , $a, b, d \in \Gamma$, $c, e \in \Sigma$, $p, q \in K$.
Hvis $L(G) = \#\,L\,\#$ gælder der, at $L = T(M)$.

Bevis. Ved hjælp af de context free produktioner i) genererer grammatikken en streng af formen $\#\,(a_1, a_1) \ldots (a_n, a_n)\,\#$ hvorefter vi tvinges til at bruge produktion nr. ii) til at få $\#\,[q_o, a_1, a_1] \ldots (a_n, a_n)\,\#$.

Herefter simulerer grammatikken automatens bevægelser idet den opererer på førstekomponenten i hjælpesymbolerne (· , ·). Anden komponenten bruges til at huske, hvad der oprindeligt stod på denne plads. Såfremt automaten accepterer strengen $a_1 \ldots a_n$, dvs. $\exists q \in F$ og $\exists \beta \in \Gamma^* : (q_0 a_1 \ldots a_n) \vdash^* (\beta q)$ sikrer produktionerne vi) og vii), at vi slutter med $\# a_1 \ldots a_n \#$. Vi vil ikke give et mere formelt bevis, idet det skulle være klart, at konstruktionen virker. □

Definition 4.10 En LBTR $M = (K, \Sigma, \Gamma, \Delta, \delta, q_0, F)$ siges at være ϵ -fri såfremt

$$\begin{array}{l} (p,b,X,y) \in \delta(q,a) \\ \Downarrow \\ y \neq \epsilon \end{array}$$ □

Lemma 4.11 Lad $M = (K, \Sigma, \Gamma, \Delta, \delta, q_0, F)$ være en ϵ -fri LBTR. Så er Ran(M) context sensitiv.

Bevis Da M er ϵ -fri, kan vi antage, at y i $(p,b,X,y) \in \delta(q,a)$ tilhører Δ , idet vi altid v.hj.a. en ϵ -fri homomorfi, som bevarer context sensitive sprog, kan transformere tilbage til det oprindelige. Vi vil konstruere en context sensitiv grammatik for Ran (M). Lad $\bar{\Delta} = \{\bar{A} \mid A \in \Delta\}$ og sæt $G = (\Gamma \cup \{\Gamma \times \Delta\} \cup \{K \times \Gamma\} \cup \bar{\Delta}, \Delta, P, S, \#)$ hvor P er følgende produktioner

i) $S \to Sa$, $S \to a$ for alle $a \in \Sigma$

ii) $\# a \to \# [q_0, a]$ for alle $a \in \Sigma$

iii) $(p,b,L,A) \in \delta(q,a) \Leftrightarrow (d,B)[q,a] \to BA[p,d]\, b$

iv) $(p,b,N,A) \in \delta(q,a) \Leftrightarrow [q,a] \to A[p,b]$

v) $(p,b,R,A) \in \delta(q,a) \Leftrightarrow [q,a]\, d \to (b,A)[p,d]$

vi) $(a,A)\, B \to A\, (a,B)$

vii) $(p,b,R,A) \in \delta(q,a) \Leftrightarrow [q,a]\, \# \to \bar{A}\, \#$ for $p \in F$

viii) $(a, A)\,\bar{B} \to \bar{A}\,B$

$A\,\bar{B} \to \bar{A}\,B$

ix) $\#\,\bar{A} \to \#\,A$

hvor $a, b, d \in \Gamma$ og $A, B \in \Delta$

Med $L(G) = \# L \#$ er $L = \mathrm{Ran}\,(M)$. Bevis herfor findes i Ginsburg og Rose [1966], og vi vil ikke gå ind på det her. Argumentet er stort set det samme som i lemma 4.9. □

En fornemmelse af, hvordan konstruktionen virker, fås af følgende eksempel.

Eksempel 4.12 Næsten ethvert eksempel, der har noget med context sensitive sprog at gøre, har traditionelt også noget med sproget $\{a^n\ b^n\ c^n \mid n \geq 1\}$ at gøre, og vi skal da også nok vare os for at gøre undtagelser her.

Betragt følgende LBTR $M = (K, \Sigma, \Gamma, \Delta, \delta, q_o, F)$ hvor

$K = \{q_o, q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, b, c, \bar{a}, \bar{b}, \bar{c}\}$

$\Delta = \{1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

$F = \{q_4\}$

og δ er defineret ved

I $\delta(q_o, a) = (q_1, \bar{a}, R, 1)$

II $\delta(q_1, a) = (q_1, a, R, 2)$

III $\delta(q_1, b) = (q_2, \bar{b}, R, 3)$

IV $\delta(q_1, \bar{b}) = (q_1, \bar{b}, R, 4)$

V $\delta(q_2, b) = (q_2, b, R, 5)$

VI $\delta(q_2, \bar{c}) = (q_2, \bar{c}, R, 6)$

VII $\delta(q_2, c) = (q_3, \bar{c}, L, 7)$

VIII $\delta(q_3, \begin{Bmatrix} b \\ \bar{b} \\ a \\ \bar{c} \end{Bmatrix}) = (q_3, L, \begin{Bmatrix} b \\ \bar{b} \\ a \\ \bar{c} \end{Bmatrix}, \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix})$

IX $\delta(q_3, \bar{a}) = (q_o, \bar{a}, R, 8)$

X $\delta(q_o, \bar{b}) = (q_4, \bar{b}, R, 9)$

XI $\delta(q_4, \begin{Bmatrix} \bar{b} \\ \bar{c} \end{Bmatrix}) = (q_4, \begin{Bmatrix} \bar{b} \\ \bar{c} \end{Bmatrix}, R, \begin{Bmatrix} E \\ F \end{Bmatrix})$

Det er nemt at indse, at Dom (M) = $\{a^n\ b^n\ c^n \mid n \geq 1\}$ og at M virker i overensstemmelse med følgende algoritme

1: if symbol = a then $\downarrow \bar{a}$ else if symbol = $\bar{b}$ then goto 2 ; $\rightarrow$;
while symbol = a $\vee$ $\bar{b}$ do $\rightarrow$;
if symbol = b then $\downarrow \bar{b}$; $\rightarrow$;
while symbol = b $\vee$ $\bar{c}$ do $\rightarrow$;
if symbol = c then $\downarrow \bar{c}$; $\leftarrow$;
while symbol = b $\vee$ $\bar{b}$ $\vee$ a $\vee$ $\bar{c}$ do $\leftarrow$;
if symbol = $\bar{a}$ then $\rightarrow$;
goto 1 ;
2: while symbol = $\bar{b}$ $\vee$ $\bar{c}$ do $\rightarrow$;

symbol betegner det symbol, M i øjeblikket læser; $\downarrow x$ betyder, at x skrives, $\rightarrow$, $\leftarrow$ betegner et ryk hhv. til højre og til venstre. Der er ikke taget hensyn til output.

Lemma 4.11 giver følgende grammatik for Ran (M) :

i) $S \rightarrow Sa,\ S \rightarrow Sb,\ S \rightarrow Sc$
$S \rightarrow a,\ S \rightarrow b,\ S \rightarrow c$

ii) $\#\, a \rightarrow \#[q_o, a],\ \#\, b \rightarrow \#[q_o, b],\ \#\, c \rightarrow \#[q_o, c]$

I $[q_o, a]\ d \rightarrow (\bar{a}, 1)\ [q_1, d]$
II $[q_1, a]\ d \rightarrow (a, 2)\ [q_1, d]$
III $[q_1, b]\ d \rightarrow (\bar{b}, 3)\ [q_2, d]$
IV $[q_1, \bar{b}]\ d \rightarrow (\bar{b}, 4)\ [q_1, d]$
V $[q_2, b]\ d \rightarrow (b, 5)\ [q_2, d]$
VI $[q_2, \bar{c}]\ d \rightarrow (\bar{c}, 6)\ [q_2, d]$
VII $(d, Z)\ [q_2, c] \rightarrow Z\ 7\ [q_3, d]\ \bar{c}$

VIII $\quad (d,Z)\,\left[q_3, \left\{\begin{matrix} b \\ \bar{b} \\ a \\ \bar{c} \end{matrix}\right\}\right] \to Z \left\{\begin{matrix} A \\ B \\ C \\ D \end{matrix}\right\}, \; [q_3, d] \; \left\{\begin{matrix} b \\ \bar{b} \\ a \\ \bar{c} \end{matrix}\right\}$

IX $\quad [q_3, \bar{a}]\; d \to (\bar{a}, 8)\; [q_o, d]$

X $\quad [q_o, \bar{b}]\; d \to (\bar{b}, 9)\; [q_4, d]$

XI $\quad \left[q_4, \left\{\begin{matrix} \bar{b} \\ \bar{c} \end{matrix}\right\}\right] d \to \left(\left\{\begin{matrix} \bar{b} \\ \bar{c} \end{matrix}\right\}, \left\{\begin{matrix} E \\ F \end{matrix}\right\} \right) [q_4, d]$

vi) $\quad (d, Z)\; Y \to Z\; (d, Y)$

vii) $\quad [q_o, \bar{b}]\; \# \to \bar{9}\; \#, \quad [q_4, \bar{b}]\# \to \bar{E}\; \#, \quad [q_4, \bar{c}]\; \# \to \bar{F}\; \#$

viii) $\quad (d, Z)\; \bar{Y} \to \bar{Z}\; Y$

$\qquad Z\; \bar{Y} \to \bar{Z}\; Y$

ix) $\quad \#\; \bar{Z} \to \#Z$

hvor $d \in \Gamma$ og $\mathbf{Z}$, $Y \in \Delta$

Lad os prøve at betragte en derivation af output fra ordet abc – ethvert større eksempel svulmer voldsomt op –.

$\# S \#$

(i)*

$\#\, abc \#$

(ii)

$\# [q_o, a]\; bc \#$

I

$\#(\bar{a}, 1)\; [q_1, b]\; c \#$

III

$\# (\bar{a}, 1)\; (\bar{b}, 3)\; [q_2, c]\; \#$

VII

$\# (\bar{a}, 1)\; 37\; [q_3, \bar{b}]\; \bar{c}\; \#$

(vi)*

$\# 13\; (\bar{a}, 7)\; [q_3, \bar{b}]\; \bar{c}\; \#$

VIII

$\# 137B\; [q_3, \bar{a}]\; \bar{b}\bar{c}\; \#$

IX

$\# 137B\; (\bar{a}, 8)\; [q_o, \bar{b}]\; \bar{c}\; \#$

X

$\# 137B\; (\bar{a}, 8)\; (\bar{b}, 9)\; [q_4, \bar{c}]\; \#$

(vii)

$\# 137B\; (\bar{a}, 8)\; (\bar{b}, 9)\; \bar{F}\; \#$

(viii)

$\# 137B\; (\bar{a}, 8)\; \bar{9}\; F\; \#$

(viii)

$\# 137B\; \bar{8}\; 9\; F\; \#$

(viii)*

$\# \bar{1}37B\; 8\; 9\; F\; \#$

ix)

$\# 137B\; 8\; 9\; F\; \#$

Hver streng i eksemplet deriveres af den, der står ovenover under en eller flere (*) anvendelser af den eller de regler, hvis nummer står mellem strengene. □

Det fremgår, at de to grammatikker i lemma 4.9 og lemma 4.11 "strukturelt" set er ens. Dette anvendes nu til at vise følgende

<u>Lemma 4.13</u> Lad $M = (K, \Sigma, \Gamma, \Delta, \delta, q_o, F)$ være en ϵ-fri LBTR. Sproget

$$D = \{ wy \mid (w, y) \in Gr(M) \}$$

er da context sensitivt.

<u>Bevis</u> Konstruer følgende context sensitive grammatik $G = (\{\Gamma \times \Sigma\} \cup \{\Gamma \times \Sigma \times \Delta\} \cup \{K \times \Gamma \times \Sigma\} \cup \bar{\Delta} \cup \hat{\Sigma}, \Sigma \cup \Delta, P, S, \#)$, hvor vi har sat $\bar{\Delta} = \{\bar{A} \mid A \in \Delta\}$, $\hat{\Sigma} = \{\hat{c} \mid c \in \Sigma\}$ og hvor P er følgnede produktioner

i) $S \to S\,(a, a)$; $S \to (a, a)$ for alle $a \in \Sigma$

ii) $\#\,(a, a) \to \#\,[q_o, a, a]$ for alle $a \in \Sigma$

iii) $(p, b, L, A) \in \delta(q, a) \Updownarrow (d, e, B)\,[q, a, c] \to BA\,[p, d, e]\,(b, c)$

iv) $(p, b, N, A) \in \delta(q, a) \Updownarrow [q, a, c] \to A\,[p, b, c]$

v) $(p, b, R, A) \in \delta(q, a) \Updownarrow [q, a, c]\,(d, e) \to (b, c, A)\,[p, d, e]$

vi) $(a, c, A)\,B \to A\,(a, c, B)$

vii) $(p, b, R, A) \in \delta(q, a) \Updownarrow [q, a, c]\,\# \to \bar{A}\,c\,\#$ for $p \in F$

viii) $(a, c, A)\,\bar{B} \to \bar{A}\,B\,\hat{c}$

ix) $\#\,A \to \#\,A$

x) $\hat{c}\,A \to A\,\hat{c}$

xi) $\hat{c}\,e \to c\,e$

Her er $a, b, d \in \Gamma$, c, $e \in \Sigma$ og $A, B \in \Delta$.

At vi med $L(G) = \# L \#$ har $L = D$ er en direkte konsekvens af lemma 4.9 og 4.11 samt den bemærkning at produktionerne x) og xi) sikrer os,at alle elementer i Σ "flyttes helt hen til højre", inden de konverteres til terminale symboler. □

Bevis for sætning 4.8 Konstruer en ϵ-fri LBTR $M_1 = (K, \Sigma, \Gamma, \Delta \cup \{\Lambda\}, \delta_1, q_o, F)$ ved at sætte

$$\begin{array}{l} (p, b, X, \Lambda) \in \delta_1(q, a) \\ \Updownarrow \\ (p, b, X, \epsilon) \in \delta(q, a) \end{array}$$

Iflg. lemma 4.13 er

$$D = \{ wy \mid (w, y) \in Gr(M) \}$$

context sensitivt. Da vi uden indskrænkning kan antage, at $\Delta \cap \Sigma = \emptyset$, kan vi definere homomorfierne h_1, h_2 ved

$$h_1(\Lambda) = h_2(\Lambda) = \epsilon$$

$$h_1(c) = \begin{cases} c \text{ hvis } c \in \Sigma \\ \epsilon \text{ ellers} \end{cases}$$

$$h_2(c) = \begin{cases} \epsilon \text{ hvis } c \in \Sigma \\ c \text{ ellers} \end{cases}$$

hvorefter det er klart, at

$$Gr(M) = \{ (h_1(w), h_2(w)) \mid w \in D \}$$ □

I forlængelse af resultaterne fra afsnit 3 er det nu rimeligt at spørge, om sætning 4.8 også gælder "den anden vej", dvs. om der for et givet $L \in CS$ og to givne homomorfier h_1, h_2 findes en LBTR M således, at $Gr(M) = \{ (h_1(w), h_2(w)) \mid w \in L \}$.

Ved hjælp af lemma 2.3 kan vi straks sige, at dette ikke gælder i almindelighed. Lad nemlig L være et sprog, som ikke er context sensitivt og konstruer L_1 som i lemma 2.3. Definer homomorfierne h_1, h_2 ved

$$h_1(a) = \begin{cases} \epsilon & \text{hvis } a = \# \\ a & \text{ellers} \end{cases}$$

$$h_2(a) = \begin{cases} \# & \text{hvis } a = \# \\ \epsilon & \text{ellers} \end{cases}$$

Nu er $h_1(L_1) = L \notin CS$, og da $\mathrm{Dom}(M) \in CS$ for enhver LBTR M er vi færdige.

Næste spørgsmål er nu, om det er muligt at afgøre for givet $L \in CS$ og givne homomorfier h_1, h_2 hvorvidt der findes en LBTR med $\{(h_1(w), h_2(w)) \mid w \in L\}$ som graf.

Svaret gives af følgende

<u>Sætning 4.14</u> Spørgsmålet om der for givet $L \in CS$ og givne homomorfier h_1, h_2 findes en LBTR M med

$$\mathrm{Gr}(M) = \{(h_1(w), h_2(w)) \mid w \in L\}$$

er rekursivt uløseligt.

<u>Bevis</u> Lad $L \subseteq \Sigma^*$ være context sensitivt, og lad os betragte det specielle tilfælde, hvor $h_2(a) = \epsilon$ for alle $a \in \Sigma$. Der gælder da, at der findes en LBTR M med de ønskede egenskaber hvis og kun hvis $h_1(L) \in CS$. Hvis vores spørgsmål derfor var rekursivt løseligt, ville det være rekursivt løseligt, om det homomorfe billede af et context sensitivt sprog igen er context sensitivt. Iflg. lemma 2.3 implicerer det imidlertid, at det er rekursivt løseligt, om et givet Turing sprog er context sensitivt. Beviset følger herefter af lemma 4.15. □

Lemma 4. 15 Spørgsmålet, om et givet Turing sprog er context sensitivt, er rekursivt uløseligt.

Bevis Lad $T = \{x_1, \ldots, x_k\}$, $U = \{w_1, \ldots, w_k\}$ være PCP-lister over alfabetet Σ, og lad $G = (V_N, \Sigma, P, S)$ være en grammatik, således at $\epsilon \notin L(G)$ og $L(G) \notin CS$. Betragt følgende grammatik $G_1 = (\{S_1, A, B, C\}, \Sigma \cup \{\#\}, P_1, S_1)$ hvor P_1 er produktionerne

$$
\begin{array}{ll}
S_1 \to \# A \# & \\
A \to aAa & \text{for alle } a \in \Sigma \\
A \to B & \\
w_i\, B\, x_i^R \to B & \\
\# B \# \to C & \\
C \to a\, C & \text{for alle } a \in \Sigma \\
C \to a &
\end{array}
$$

(Hvis $x = a_1 \ldots a_n$ er $x^R = a_n \ldots a_1$).

Lad endelig G' være grammatikken $G' = (\{S_1, S, A, B, C\} \cup V_N, \Sigma \cup \{\#\}, P_1 \cup P_2 \cup \{S' \to S, S' \to S_1\}, S')$. Det er klart, at $L(G') = L(G) \cup L(G_1)$. Af konstruktionen af G_1 følger der nu, at

$$
L(G_1) = \begin{cases} \emptyset \text{ hvis } PCP(T,U) = 0 \\ \Sigma^+ \quad PCP(T,U) = 1 \end{cases}
$$

men det betyder åbenbart, at

$$
\begin{array}{l} L(G') \text{ er context sensitivt} \\ \Updownarrow \\ PCP(T,U) = 1 \end{array}
$$

Det eneste, vi mangler at vise, er at vi altid effektivt kan konstruere en grammatik G, således at L(G) ikke er context sensitiv.

Lad $\{\varphi_x\}$ betegne en nummerering af de partielle rekursive funktioner og lad a være et element i Σ. Sproget

$$
L = \{a^n \mid n \geq 1,\ \varphi_n(n) \text{ konvergent}\}
$$

er ikke rekursivt, hvilket følger af uløseligheden af Halting problemet.

På den anden side er det simpelt at konstruere en Turing Acceptor, der accepterer L, det er nemlig den universelle Turing Maskine, der givet n udregner $\varphi(n)$ og accepterer hvis og kun hvis den standser. Givet vor Turing Acceptor T kan vi nu anvende den algoritme, der findes i Hopcroft og Ullmann [1969] til konstruktion af en grammatik for det sprog, der accepteres af T. □

Lad os ligesom i afsnit 3 slutte dette afsnit med at karakterisere oversættelser defineret af LBTRier ved hjælp af grammatikker.

Definition 4. 16 En context sensitiv grammatik med regulær output (CSRO) er et system $G = (V_{N_1}, V_{N_2}, V_{T_1}, V_{T_2}, P, (S, S'))$ hvor

i) V_{N_1}, V_{N_2} er alfabeter af hjælpesymboler,
   V_{T_1}, V_{T_2} – – – terminale symboler
   med $V_{N_i} \cap V_{T_i} = \emptyset \quad i = 1, 2.$

ii) $(S, S') \in V_{N_1} \times V_{N_2}$ er startsymbolet

iii) P er en mængde af produktioner af en af følgende former

$$(\alpha, A) \to (\beta, xB)$$
$$(\alpha, A) \to (\beta, x)$$

hvor $\alpha, \beta \in V_1^+$ $|\beta| \geq |\alpha|$, $A, B \in V_{N_2}$, $x \in V_{T_2}^*$

En form er et element (α, x) hvor $\alpha \in V_1^+$, $x \in V_2^*$, Definer relationen $\Rightarrow$ mellem former ved

$$(\alpha, xA) \Rightarrow (\beta, \eta)$$

hvis vi kan skrive (α, xA) på formen $(\omega \delta \gamma, xA)$ og (β, η) på formen $(\omega \mu \gamma, x\upsilon)$ og $(\delta, A) \to (\mu, \upsilon)$ er en produktion.

Med $\overset{*}{\Rightarrow}$ som den reflexive, transitive lukning af $\Rightarrow$, definerer vi sproget genereret af G som mængden

$$L(G) = \{ (x, y) \in V_{T_1}^* \times V_{T_2}^* \mid (S, S') \overset{*}{\Rightarrow} (x, y) \} \qquad \square$$

Ovenstående er et specialtilfælde af Culik II's og Morey's [1971] definition af binær grammatik og af det sprog, der α-genereres af en sådan.

Vi har følgende sætning fra samme artikel.

Sætning 4. 17 Lad Σ, Δ være to alfabeter og lad $T \subseteq \Sigma^* \times \Delta^*$. Der findes da en LBTR M med $Gr(M) = T$ hvis og kun hvis der findes en CSRO G med $L(G) = T$.

Bevis Beviset er gennemført i alle pinlige detaljer i ovenstående artikel, hvorfor vi ikke skal komme nøjere ind på det. Ideen er meget kort den, at vi givet en LBTR M med $T = Gr(M)$ konstruerer en CSRO, hvis context sensitive del praktisk taget er den samme som grammatikken i lemma 4.9 og hvis regulære del kun indeholder et hjælpesymbol og tre forskellige produktioner $S \to S$, $S \to yS$, $S \to \varepsilon$ hvor y er en output streng fra M.

En produktion, der "stammer" fra en bevægelse, hvor automaten giver output y, har formen $(\alpha \to \beta,\ S \to yS)$, en "hjælpeproduktion" er af formen $(\alpha \to \beta,\ S \to S)$ og endelig er den sidst anvendte produktion i en derivation produktionen $(\alpha \to \beta,\ S \to \varepsilon)$.

Har vi omvendt givet en CSRO konstruerer vi en LBTR, der accepterer sproget genereret af den context sensitive del af grammatikken v.hj.a. en "top-down analyse" (jfr. Hopcroft og Ullmann [1969] Theorem 8. 1) og som samtidig anvender den regulære del af grammatikken til at generere det korrekte output. $\square$

Vi skal ikke her komme yderligere und på oversættelser defineret af LBTR'er, men slutter med at henvise til Culik II og Morey, hvor specielt sammenhængen mellem oversættelser og binære grammatikker er behandlet i dybden.

AFSNIT 5

TURING ACCEPTORER OG TURING OVERSÆTTERE

Dette afsnit vil på afgørende vis adskille sig fra de øvrige, idet vi her vil bruge en helt anden type argumenter. Medens de hidtige beviser har været formelle og stringente i den forstand, at vi f. eks. har konstrueret grammatikker og automater i alle tekniske detaljer, vil vi nu slå ind på "de store armbevægelsers metode", eller hvad der er en mere korrekt betegnelse, vi vil anvende Church's Thesis. Denne siger som bekendt, at en Turing maskine kan gøre alt det, som vi kan give en effektiv beskrivelse af. Derfor vil beviser i høj grad blive givet i form af flow-charts, hvis elementer beskriver en beregning, som intuitivt er effektiv.

Først en formel definition.

Definition 5.1 En Turing acceptor (TA) er et system $M = (K, \Sigma, \Gamma, \delta, q_o, F)$ hvor $K, \Sigma, \Gamma, \delta, q_o, F$ har samme betydning som i def. 4.1 og der yderligere gælder, at der findes et specielt symbol $B \in \Gamma \setminus \Sigma$ – B er den blanke –.

Indfør relationen $\vdash^*$ på $\Gamma^* K \Gamma^*$ ved for $u, v \in \Gamma^*$, $c \in \Gamma$ at sætte

i) $upav \vdash uqbv \iff \delta(p,a) \ni (q,b,N)$ $\qquad$ $up \vdash uqb \iff \delta(p,B) \ni (q,b,N)$

ii) $upav \vdash ubqv \iff \delta(p,a) \ni (q,b,R)$ $\qquad$ $up \vdash ubq \iff \delta(p,B) \ni (q,b,R)$

iii) $$\left.\begin{array}{l} ucpav \vdash uqcbv \\ pav \vdash qBav \end{array}\right\} \iff \delta(p,a) \ni (q,b,L)$$

$$\left.\begin{array}{l} ucp \vdash uqcb \\ p \vdash qBb \end{array}\right\} \iff \delta(p,B) \ni (q,b,L)$$

Med $\vdash^*$ som den reflexive, transitive lukning af $\vdash$ er sproget accepteret af M mængden

$$T(M) = \{ w \mid \exists\, q \in F, \alpha, \beta \in \Gamma^*: q_0 w \vdash^* \alpha q \beta \} \qquad \square$$

Det følger af definitionen, at hvis $q \in F$ kan vi uden indskrænkning antage, at M standser, dvs. at $\delta(q, a)$ er udefineret for alle $a \in \Gamma$.

Det er velkendt, at de mere udbyggede modeller af TA'er, dvs. maskiner med flere tapes af flere dimensioner og med flere læse-skrive hoveder, alle er ækvivalente med ovenstående model.

På basis af def. 5.1 kan vi nu definere begrebet en Turing oversætter (TTR) ved at tilføje et outputalfabet Δ, pille lidt ved definitionerne af δ og $\vdash$, og dermed nå frem til at definere grafen for en TTR $M = (K, \Sigma, \Gamma, \Delta, \delta, q_0, F)$ som

$$Gr(M) = \{ (x, y) \in \Sigma^* \times \Delta^* \mid \exists\, q \in F, \alpha, \beta \in \Gamma^*: (q_0, w, \varepsilon) \vdash^* (\alpha q \beta, \varepsilon, y) \}$$

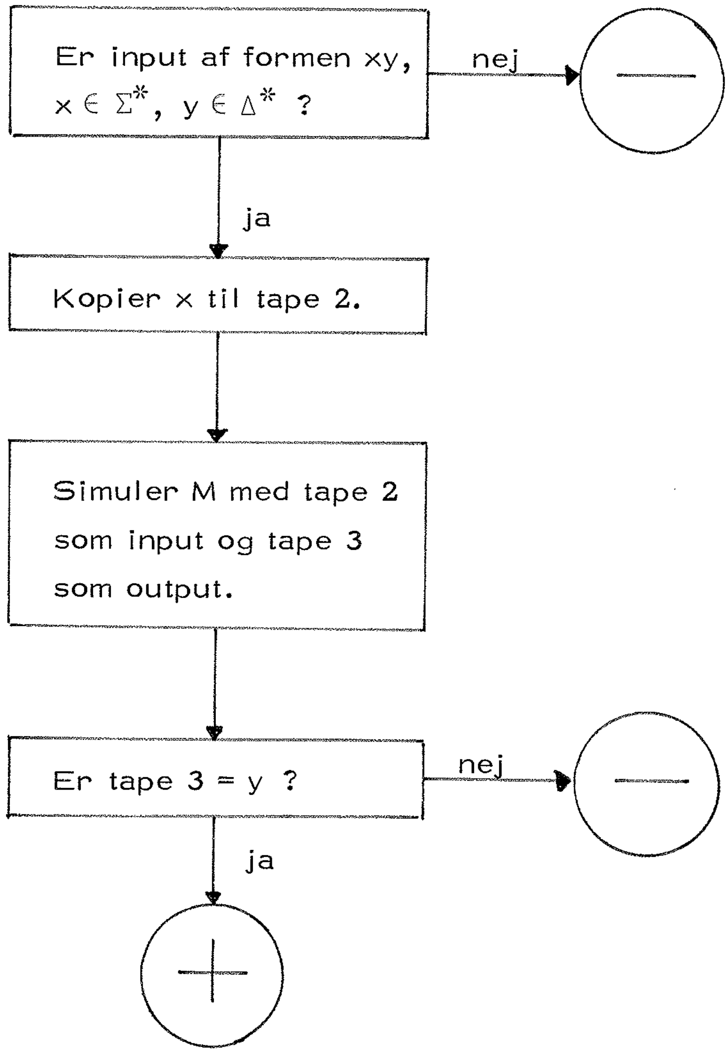
De ovenfor beskrevne skridt er fuldstændig parallelle til def. 4.3 og def. 4.4, og det skulle være indlysende, hvordan det går.

Lad os nu vise følgende parallel til corollar 3.11.

Sætning 5.2 Lad Σ, Δ være to alfabeter med $\Sigma \cap \Delta = \emptyset$ og lad $T \subseteq \Sigma^* \times \Delta^*$. Der findes da en TTR M med $T = Gr(M)$ hvis og kun hvis der findes et Turing sprog L og to homomorfier h_1, h_2 således at

$$T = \{ (h_1(w), h_2(w)) \mid w \in L \}$$

Bevis Lad $M = (K, \Sigma, \Gamma, \Delta, \delta, q_0, F)$ være en TTR med $T = Gr(M)$. Konstruer en TA M_1 på følgende måde. M_1 har tre tapes, inputalfabet $\Sigma \cup \Delta$ og virker i overensstemmelse med følgende diagram idet tape 1 bruges som inputtape.



Bemærk at det er muligt, at vor maskine aldrig bliver færdig med at simulere M. − angiver en ikke-accepttilstand, medens + angiver en accepttilstand.

Lad L være lig med $T(M_1)$ og definer homomorfierne h_1 og h_2 ved

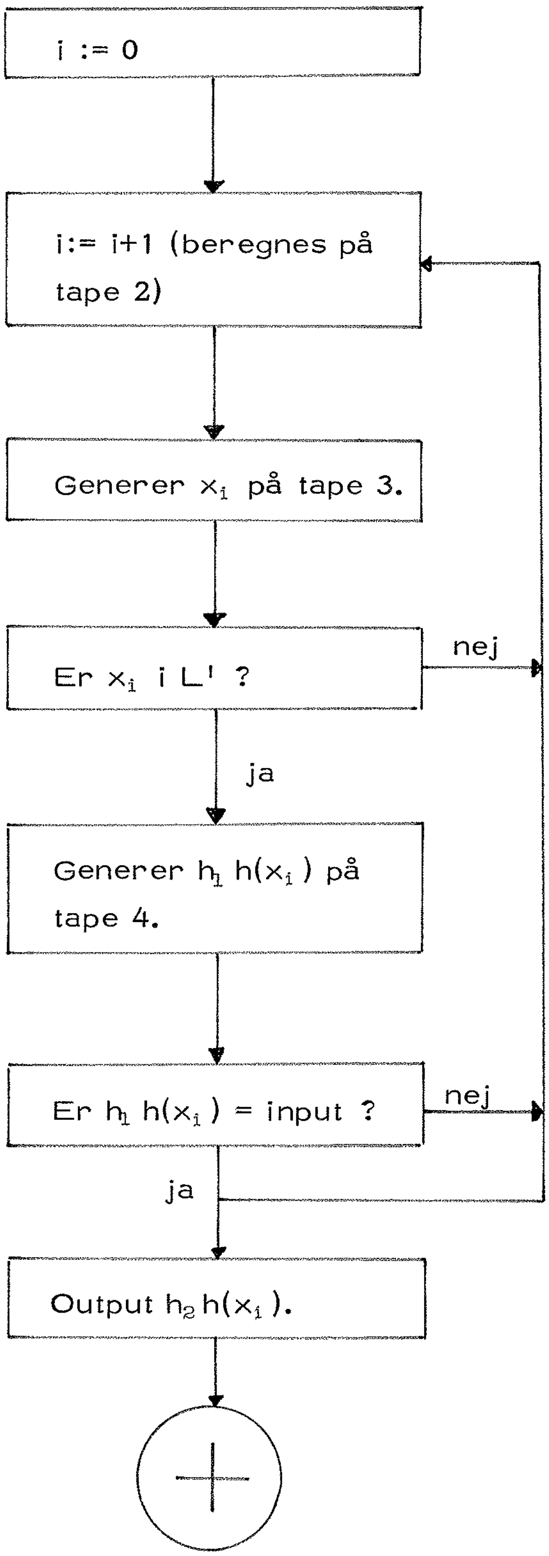
$$\text{for } a \in \Sigma \text{ er } h_1(a) = a,\ h_2(a) = \varepsilon$$
$$\text{for } a \in \Delta \text{ er } h_1(a) = \varepsilon,\ h_2(a) = a$$

Det er klart at $T = \{ (h_1(w), h_2(w)) \mid w \in L \}$.

Lad omvendt $T = \{ (h_1(w), h_2(w)) \mid w \in L \}$, hvor $L \in RE$ med $L \subseteq \Sigma'^*$. Vi anvender atter lemma 2.3 til at opnå $L' \in CS$, $L' \subseteq (\Sigma' \cup \{\#\})^*$, $h(L') = L$, hvor homomorfien h er defineret ved $h(\#) = \epsilon$, $h(a) = a$ for $a \in \Sigma$. Vi har nu at

$$T = \{ (h_1\, h(w), h_2\, h(w)) \mid w \in L' \}$$

Lad $\{ x_i \}$ være en nummerering af strengene i $(\Sigma' \cup \{\#\})^*$ og konstruer følgende TTR med inputalfabet Σ, outputalfabet Δ og 4 tapes med tape 1 som input.



Bemærk at det efter svaret ja i næstsidste kasse er to muligheder for TTR'en. Dette skyldes, at vi må bevare muligheden for at få flere forskellige output fra samme input. Den ekstra detalje med at indføre $L' \in CS$ skyldes, at vi i kasse nr. 4 skal have et spørgsmål, som er rekursivt løseligt. □

Lad os som kommentar til ovenstående sætning anføre, at den også kunne have været vist på følgende måde. Turing sprogene er en full AFL $\mathcal{T}$, hvorfor der iflg. sætning 2.6 findes en AFTR $\mathfrak{A}$ så $\mathcal{L}(\mathfrak{A}) = \mathcal{T}$. Anvend sætning 3.6 på $\mathfrak{A}$ og vi er tilsyneladende hjemme. Her er imidlertid en ting at være opmærksom på, og det er kravet om, at vore automater i en AFTR er <u>envejs</u> automater. Ovenstående klasse $\mathfrak{A}$ er altså en klasse af envejsautomater, som ganske vist er ækvivalent med klassen af Turing maskiner i den forstand, at de accepterer de samme sprog, men som strukturelt set ikke behøver at ligne de klassiske Turing maskiner. Dette er grunden til, at vi har anvendt den mere direkte måde at angribe problemet på.

Lad os igen slutte med at angive sammenhængen mellem oversættelser og grammatikker. Ved at ændre definitionen af en context sensitiv grammatik med regulær output (CSRO i def. 4.16) således at vi i en produktion $(\alpha, A) \to (\beta, xB)$ eller $(\alpha, A) \to (\beta, x)$ ikke længere kræver at $|\beta| \geq |\alpha|$ får vi, hvad vi kan kalde en Turing grammatik med regulær output (RERO). Det er ikke svært at vise følgende sætning, hvormed vi slutter behandlingen af TTR'er.

<u>Sætning 5.3</u> Lad Σ, Δ være to alfabeter og lad $T \subseteq \Sigma^* \times \Delta^*$. Der findes da en TTR M med $Gr(M) = T$ hvis og kun hvis der findes en RERO G med $T = L(G)$. □

AFSNIT 6

KONKLUSION OG AFSLUTNING

Hvis vi anvender de sædvanlige typebetegnelser for Chomsky-hierarkiet - dvs. et type 1 sprog er context sensitivt, en type 2 automat er en PDA osv. - har vi i de foregående afsnit vist, at givet en oversætter M af type i, findes der et sprog L af type i og to homomorfier h_1 , h_2 således at

$$Gr(M) = \{ (h_1(w), h_2(w)) \mid w \in L \}$$

Bortset fra tilfældet i=1 har vi vist at denne sætning også gælder "den anden vej" ligesom vi har demonstreret at i tilfældet i=1 er det rekursivt uløseligt, hvorvidt dette sidste er opfyldt. Endelig har vi i hvert tilfælde angivet en karakterisation for graferne af oversætterne v.hj.a. "par af grammatikker".

I spørgsmålet om "anvendeligheden" af den præsenterede teori er det nok først og fremmest push-down oversætterne der kommer i betragtning. Lad os prøve at anvende en af vore sætninger på et problem, der optræder i forbindelse med syntaxdirigeret oversættelse. Aho og Ullman [1971] anfører et eksempel som i alt væsentligt er følgende:

Betragt en PASCAL-lignende produktion som

<for statement>::= <u>for</u> <assignmemt> <u>to</u> <integer> <u>do</u> <statement>

Et eksempel på et <for statement> kan være

<u>for</u> I ← J + 1 <u>to</u> N <u>do</u> B ← B + A[J]

En naturlig kode at producere vil være

1. Udfør assignmentet
2. Test om I ≤ N
3. Hvis ikke hoppes der ud

4. Udfør statementet
5. Sæt $I \leftarrow I + 1$
6. Gå tilbage til 2

Vi kan nu spørge, om denne oversættelse kan udføres af en PDTR. En nødvendig og tilstrækkelig betingelse herfor er iflg. sætning 3.13 at den oversatte kode kan beskrives af en produktion af formen

<for statement>::= t_1 <assignment> t_2<integer> t_3<statement> t_4 (*)

hvor t_i for $1 \leq i \leq 4$ er terminale strenge.

Vi kan imidlertid med det samme se, at vi i vor kode får brug for to "oversættelser" af <assignment>, nemlig dels den kode, der udfører $I \leftarrow J + 1$, og dels en kode, der kan levere I's reference, som vi skal bruge i 2 og 5. Da <assignment> i (*) imidlertid kun kan have en "værdi", kan ovenstående oversættelse ikke defineres af en PDTR.

Lad os prøve at omskrive produktionen til

<for statement>::=<u>for</u> <identifier> ← <integer expression> <u>to</u>
<integer> <u>do</u> <statement>

Lad os vise at vi nu faktisk kan definere oversættelsen v.hj.a. en PDTR ved at anføre en oversættelse til GIER's assemblerkode SLIP. Vi skal altså vise, at vi kan oversætte strengen

<u>for</u> I ← E <u>to</u> N <u>do</u> S

Lad os antage, at den semantiske værdi af I er en stump kode, der anbringer adressen på den aktuelle identifier i R-registerets adressedel. Den semantiske værdi af E hhv. N er den kode, der udregner den aktuelle værdi af E hhv. N og anbringer denne i R og endelig er den semantiske værdi af S den kode, der udregner S.

Vi kan nu skrive vor semantiske produktion af formen

$$\text{<for statement>} ::= t_1\ I\ t_2\ E\ t_3\ N\ t_4\ S\ t_5$$

på følgende måde

<for statement>::= I ga re1 V; e1: qq; E gr(re1); e2: N
sr(re1); hv re3 LT; S arn(re1) V; e4: qq 1. 39;
ar re4; gr(re1); hv re2; e3: qq;.

hvor vi har brugt ; som skilletegn mellem SLIP-instruktioner.

Et andet spørgsmål, der melder sig i forbindelse med praktiske anvendelser, er spørgsmålet om at karakterisere de oversættelser, der defineres af deterministiske push-down automater.

En SSDTS $G = (V, \Sigma, \Delta, R, S)$ (jfr. def. 3. 12) kaldes omvendt polsk, såfremt der for enhver form (α, β) gælder at $\beta \in V^* \Sigma^*$. G siges at være LR(k) såfremt grammatikken $G' = (V, \Sigma, P, S)$ er en LR(k)-grammatik. Her er P produktionerne $A \to \alpha$, hvor $A \to (\alpha, \beta)$ er en regel i R.

Lewis og Stearns [1968] indeholder nu følgende resultat.

<u>Sætning 6. 1</u> Lad Σ, Δ være to alfabeter og lad $T \subseteq \Sigma^* \times \Delta^*$. Der findes da en deterministisk push-down oversætter M med $Gr(M) = T$ hvis og kun hvis der findes en omvendt polsk LR(k) SSDTS $G = (V, \Sigma, \Delta, R, S)$ således at $L(G) = T$. □

Beviset for sætning 6. 1 er konstruktivt, og sætningen har følgelig anvendelser i forbindelse med konstruktion af LR(k)-parsere og LR(k)-parser-generatorer.

Lad os alutte med at anføre, at der ikke findes nogen "homomorfi-karakterisation" af oversættelser defineret af deterministiske PDA'er. Et forsøg på at gøre det vil utvivlsomt medføre overvejelser af samme art som for LBA'erne i afsnit 4. Det springende punkt er igen, at deterministiske sprog ikke er lukket over for homomorfi.

Referencer:

Aho and Ullman [1969]: "Syntax Directed Translations and the Pushdown Assembler". J. Comp. Syst. Sci. 3, (37-56)

Aho and Ullman [1969a]: "Properties of Syntax Directed Translations." J. Comp. Syst. Sci. 3, (319-334)

Aho and Ullman [1971]: "Translations on a Context Free Grammar." Inf. and Control 19, (439-475)

Culik II and Morey [1971]: "Formal Schemes for Language Translations." Intern. J. Computer Math. A, 3, (17-48)

Ginsburg and Greibach [1969]: "Abstract Families of Languages." Mem. of the Amer. Math. Soc., 87, (1-32)

Ginsburg and Rose [1966]: "Preservation of Languages by Transducers." Inf. and Control, 9, (153-176)

Ginsburg and Rose [1968]: "A note on perservation of Languages by Transducers." Inf. and Control, 12, (549-552)

Hopcroft and Ullman [1969]: "Formal Languages and their Relation to Automata." Addison-Wesley Inc.

Ibarra [1971]: "Characterizations of Transductions Defined by Abstract Families of Transducers." J. Math. Syst. Th. 5, 3, (271-281)

Kuroda [1964]: "Classes of Languages and Linear-Bounded Automata." Inf. and Control (207-223)

Landweber [1963]: "Three Theorems on Phrase Structure Grammars of Type 1." Inf. and Control (131-136)

Lewis and Stearns [1968]: "Syntax-Directed Transduction." JACM, 15, 3 (465-468)