TECHNISCHE HOGESCHOOL EINDHOVEN
Afdeling Algemene Wetenschappen
Onderafdeling der Wiskunde

TOEGEPASTE LOGICA II

van

Prof. Dr. W. Peremans

Lentetrimester 1985

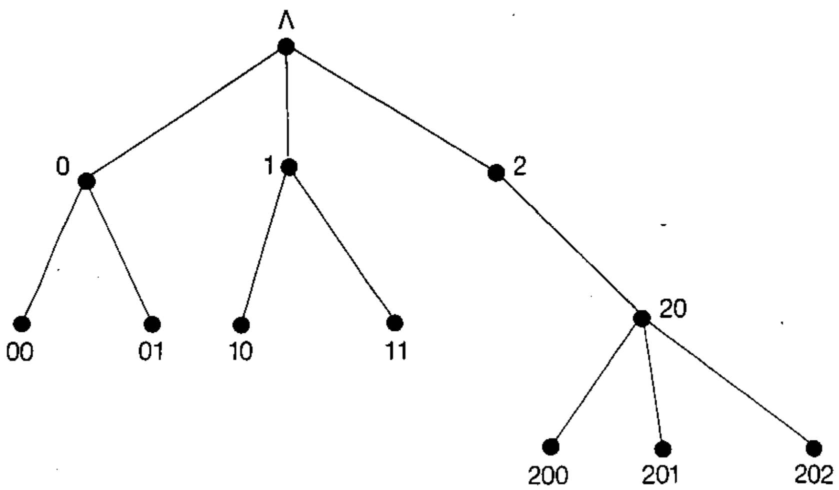
Technische Hogeschool
Eindhoven

Dictaatnummer 2.357.0
Prijs f.5,50

Onderafdeling der Wiskunde en Informatica

Toegepaste logica II

Syllabus bij het college van prof.dr. W. Peremans
(28850)



Inhoudsbeschrijving
TOEGEPASTE LOGICA II
W. Peremans
Lentetrimester 1985

JdG, 31 Juli 2005

TECHNISCHE HOGESCHOOL EINDHOVEN

Onderafdeling der Wiskunde en Informatica

TOEGEPASTE LOGICA II

Syllabus bij het college van

Prof.dr. W. Peremans

(28850)

Lentetrimester 1985

HOOFDSTUK 0. INLEIDING

§0. PRODUKTIES

Het college Toegepaste logica II is geen vervolg op het college Toegepaste logica I. Weliswaar wordt op bescheiden schaal van resultaten uit Toegepaste logica I gebruik gemaakt, maar dat hoeft geen beletsel te zijn om het college zonder kennis van Toegepaste logica I te volgen. Het college is ook minder 'fundamenteel' dan Toegepaste logica I: er zal met minder schroom van resultaten uit de gangbare wiskunde gebruik worden gemaakt.

We beginnen met een beknopte bespreking van produkties in de zin van E. Post. Stel S een eindige verzameling; de elementen van S heten symbolen. We vormen eindige rijen van elementen van S, die we symboolrijen noemen. Hierbij is ook de lege rij inbegrepen, die we met Λ aanduiden. De symboolrijen inclusief Λ vormen de verzameling S^* (exclusief Λ: S^+). Op S^* is een binaire operatie gedefinieerd, die aan symboolrijen u en v de symboolrij toevoegt, die ontstaat door v achter u te schrijven; we duiden die symboolrij met uv aan en noemen de operatie concatenatie. Het is duidelijk dat concatenatie associatief is; dit drukken we uit door te zeggen dat S^+ en S^* semigroepen zijn. Verder is Λ een neutraal element ($\Lambda u = u\Lambda = u$); daarom heet S^* een monoïde. Soms worden de elementen van S letters en de elementen van S^* woorden genoemd; voor sommige toepassingen is dit echter onpraktisch. Zo stellen in de theorie van natuurlijke talen de elementen van S soms woorden en de elementen van S^* zinnen voor.

Een produktie wordt vastgelegd door 6 elementen van S^*: $g,h,k,\bar{g},\bar{h},\bar{k}$ vast gekozen. Bij deze produktie horen produktieregels:

$$gPhQk \rightarrow \bar{g}P\bar{h}Q\bar{k}, \text{ waarin } P \in S^*,\ Q \in S^* \text{ variabel.}$$

Speciale gevallen: $g = k = \bar{g} = \bar{k} = \Lambda$; dan is de produktie semi-Thue; de regel is dan $PhQ \to P\bar{h}Q$; $h = k = \bar{g} = \bar{h} = \Lambda$; dan is de produktie normaal ; de regel heeft dan de gedaante $gP \to P\bar{k}$.

Om een combinatorisch systeem te verkrijgen kiezen we een $T \subset S$ en verder één vaste symboolrij, die we axioma noemen, benevens een eindige verzameling produkties. Een afleiding in dit systeem is een rij $X_0,\dots,X_m$ van elementen van S^* zo, dat X_0 het axioma is en $X_i \to X_{i+1}$ voor $i \in \{0,\dots,m-1\}$ een produktieregel is behorende bij een der produkties van het systeem. Een element X van S^* heet afleidbaar in het systeem , als er een afleiding $X_0,\dots,X_m$ in het systeem bestaat zo, dat $X_m = X$. De taal van het systeem is de verzameling $\{X \in T^* | X \text{ afleidbaar}\}$.

De symboolrijen in de taal bevatten uitsluitend symbolen in T; daarom heten de elementen van T eindsymbolen (Engels: terminal symbols). De elementen van $S \setminus T$ heten hulpsymbolen (Engels: non-terminal symbols). Als alle produkties in een combinatorisch systeem semi-Thue (resp. normaal) zijn, dan noemen we het systeem zelf ook semi-Thue (resp. normaal).

Een eenvoudig voorbeeld van een semi-Thue systeem is het volgende:
$S = \{A,a,b\}$, $T = \{a,b\}$; produktieregels $PAQ \to PaAbQ$ en $PAQ \to PQ$; axioma A.
Voorbeeld van een afleiding: A,aAb, aaAbb, aaaAbbb, aaabbb.
De taal van het systeem is $\{a^n b^n | n \in \mathbb{N}\}$; hierin stelt u^n de symboolrij voor die ontstaat door u n maal met zichzelf te concateneren: $u^0 = \Lambda$, $u^{n+1} = u^n u$.

Een van de vragen die men zich kan stellen is, bij gegeven T, welke deelverzamelingen van T^* als taal bij een combinatorisch systeem kunnen worden verkregen. Let wel dat daarbij de keuze van $S \setminus T$ nog vrij is; men mag net zo veel hulpsymbolen toevoegen als men zelf wil. Post heeft aangetoond dat iedere taal van een combinatorisch systeem ook te krijgen is als de taal van een normaal combinatorisch systeem. Verder heeft Post combinatorische systemen gebruikt voor een formele behandeling van het begrip beslisbaarheid; in het college Toegepaste logica I zijn daarvoor andere hulpmiddelen gebezigd. Daarop gaan we hier niet verder in.

Een beroemd voorbeeld van een combinatorisch systeem is door Post geïntroduceerd en wordt in het Engels tag system genoemd. Kies S = T = {0,1}. Een rij wordt nu als volgt gewijzigd:

als met 0 begint: voorste drie schrappen en achteraan 00 toevoegen,

als met 1 begint: voorste drie schrappen en achteraan 1101 toevoegen.

Als de rij minder dan drie symbolen bevat, stopt het proces. Dit voorschrift kan worden beschreven met de volgende acht produktieregels:

000P → P00	100P → P1101
001P → P00	101P → P1101
010P → P00	110P → P1101
011P → P00	111P → P1101

De keuze van het axioma is vrij. Enkele voorbeelden van afleidingen:

1010, 01101, 0100, 000, 00; hierna stopt het.

1101, 11101, 011101, 10100, 001101, 10100; hier wordt na drie en na vijf stappen dezelfde rij verkregen. Dit heeft als gevolg dat na drie stappen het proces periodiek wordt met periode 2: de 6^e is gelijk aan de 4^e, 7^e aan de 5^e enz.

Een ingewikkelder geval van hetzelfde verschijnsel krijgen we als we starten met 10010. Dan komt er na de 16^e en na de 22^e stap dezelfde rij, namelijk 011011101110100. Na de 16^e stap is het proces periodiek met periode 6. De vraag kan worden gesteld of iedere rij tot een stoppend, dan wel periodiek proces leidt; het antwoord op deze vraag is onbekend.

§1. GRAMMATICA'S

Een speciaal geval van een combinatorisch systeem is een grammatica.

<u>Definitie grammatica</u>

Een 4-tupel $G = \langle I, I_T, Z, R \rangle$ heet een grammatica, als geldt:

I is een eindige verzameling,

$I_T \subset I$,

$Z \in I \backslash I_T$,

R is een eindige verzameling paren (u,v), waarvoor $u \in I^*$, $v \in I^*$.

Deze definitie van grammatica is afkomstig van N. Chomsky. De elementen van I heten <u>symbolen</u>, die van I_T <u>eindsymbolen</u>, die van $I \backslash I_T$ <u>hulpsymbolen</u>; we schrijven vaak $I_N := I \backslash I_T$.

De elementen van R worden <u>produkties</u> of <u>herschrijfregels</u> genoemd. Van de regel (u,v) heet u het <u>linkerlid</u>, v het <u>rechterlid</u>. Z heet het <u>startsymbool</u>. We maken de afspraak in de theorie van de grammatica's voor de eindsymbolen Griekse letters en voor de hulpsymbolen Latijnse hoofdletters te gebruiken. Symboolrijen geven we aan met Latijnse kleine letters. Uiteraard kunnen in voorbeelden en toepassingen andere keuzen, bijvoorbeeld voor eindsymbolen worden gedaan.

Als $p \in I^*$, $q \in I^*$, dan schrijven we $p \underset{G}{\Rightarrow} q$ als er r, s, u, v, alle in I^* bestaan zo, dat $p = rus$, $q = rvs$, $(u,v) \in R$. We schrijven $p \overset{*}{\underset{G}{\Rightarrow}} q$ als er een eindige rij $p_0, \ldots, p_n$ bestaat zo, dat $p_0 = p$, $p_n = q$ en $p_i \underset{G}{\Rightarrow} p_{i+1}$ voor $i \in \{0, \ldots, n-1\}$. Men zegt wel dat de relatie $\overset{*}{\underset{G}{\Rightarrow}}$ op I^* de reflexief-transitieve afsluiting is van de relatie $\underset{G}{\Rightarrow}$. Let wel dat het in de definitie

van $\overset{*}{\underset{G}{\Rightarrow}}$ toegestaan is dat $n = 0$: $p \overset{*}{\underset{G}{\Rightarrow}} p$ geldt voor alle $p \in I^*$. Als het duidelijk is op welke grammatica de relaties betrekking hebben laat men de G in de notatie wel weg: $\Rightarrow$ en $\overset{*}{\Rightarrow}$ in plaats van $\underset{G}{\Rightarrow}$ en $\overset{*}{\underset{G}{\Rightarrow}}$.
We noemen de hierboven gegeven rij $p_0,\dots,p_n$ een <u>afleiding</u> van q uit p in G.

<u>Definitie taal gegenereerd door grammatica</u>

Stel $G = \langle I, I_T, Z, R\rangle$ een grammatica. De taal L(G), gegenereerd door G wordt gedefinieerd door

$$L(G) := \{x \in I_T^* \mid Z \overset{*}{\underset{G}{\Rightarrow}} x\} .$$

Een afleiding van x uit Z wordt wel kortweg een afleiding van x genoemd.

Alvorens voorbeelden te geven behandelen we enkele begrippen met betrekking tot symboolrijen. Definities en bewijzen betreffende symboolrijen kunnen vaak gevoerd worden met <u>recurrentie langs de opbouw</u> van symboolrijen: we behandelen Λ en voor een symboolrij x en een symbool ξ gaan we uit van x en stappen over op $x\xi$. Zo kunnen we de <u>gespiegelde</u> x^T van een symboolrij x definiëren door $\Lambda^T := \Lambda$ en $(x\xi)^T := \xi x^T$ en de <u>lengte</u> $\ell(x)$ van een symboolrij door $\ell(\Lambda) := 0$ en $\ell(x\xi) := \ell(x) + 1$. Er geldt dan: $(xy)^T = y^T x^T$, $(x^T)^T = x$, $\xi^T = \xi$, $\ell(xy) = \ell(x) + \ell(y)$, $\ell(\xi) = 1$, $\ell(x^T) = \ell(x)$.

<u>Voorbeeld 1</u>. $I_T = \{\alpha,\beta\}$, $I_N = \{Z\}$, $R = \{(Z,\alpha Z\beta), (Z,\Lambda)\}$.
Dan is $L(G) = \{\alpha^n\beta^n \mid n \in \mathbb{N}\}$. (Evenals in het college Toegepaste logica I wordt in dit college het getal nul tot de natuurlijke getallen gerekend.)

<u>Voorbeeld 2</u>. $I_T = \{\alpha,\beta\}$, $I_N = \{Z\}$, $R = \{(Z,\alpha Z\alpha), (Z,\beta Z\beta), (Z,\Lambda)\}$.
Dan is $L(G) = \{xx^T \mid x \in I_T^*\}$. Men noemt een symboolrij, waarvoor geldt $x^T = x$, wel een <u>palindroom</u>. Nu bestaat L(G) precies uit de palindromen van

even lengte, want voor $y \in I_T^*$ geldt namelijk:

$$y^T = y \text{ en } \ell(y) \text{ even} \leftrightarrow \exists_{x \in I_T^*}[y = xx^T].$$

Voorbeeld 3. $I_T = \{\alpha,\beta\}$, $I_N = \{Z,A\}$,

$R = \{(Z,AZ),(A,\alpha A\beta),(A,\alpha),(Z,\Lambda)\}$. Voorbeeld van een afleiding:

$Z, AZ, AAZ, AAAZ, A\alpha A\beta AZ, \alpha\alpha A\beta AZ, \alpha\alpha\alpha\beta AZ, \alpha\alpha\alpha\beta\alpha Z, \alpha\alpha\alpha\beta\alpha$.

Opgave 1. Bepaal de taal die gegenereerd wordt door de grammatica van Voorbeeld 3.

Opgave 2. Construeer een grammatica, die de palindromen van oneven lengte genereert.

De hierboven gedefinieerde grammatica's, die door Chomsky phrase-structure grammars zijn genoemd, zijn zeer algemeen en zelfs voor vele doeleinden te algemeen. Men legt daarom aan de grammatica's beperkingen op; deze hebben vaak betrekking op de gedaante van de herschrijfregels. Chomsky heeft bepaalde beperkingen typen genoemd en deze met de nummers 0, 1, 2, 3 aangeduid; deze aanduidingen zijn gangbaar geworden. Type 0 is het algemene geval zonder beperkingen.

Wij geven nu een overzicht van een aantal beperkingen en hun consequenties, voorlopig zonder bewijzen. We nemen daarbij aan, dat de verzameling I_T van eindsymbolen vast gekozen is. Het feit dat hierop volgende definities van de keuze van I_T afhangen, brengen we niet in de notatie tot uitdrukking.
Een deelverzameling van I_T^* noemen we een taal. Talen worden met hoofdletters aangeduid. Een verzameling van talen (ook wel klasse van talen genoemd) duiden we aan met een schrijfhoofdletter. Zo zal $\mathcal{L}_0$ de klasse zijn van de talen, waarvoor een grammatica bestaat die hem genereert: de talen van type 0.

Als B een beperking voor grammatica's voorstelt, dan is $\mathcal{L}_B$ de klasse van de talen, die gegenereerd kunnen worden door een grammatica, die aan de beperking B voldoet. Bijzondere klassen van talen zijn $\mathcal{R}$, de klasse der recursieve talen en $\mathcal{RE}$, de klasse der recursief-opsombare talen. Intuïtief betekent $L \in \mathcal{R}$, dat er een effectieve methode bestaat om voor iedere $x \in I_T^*$ uit te maken of $x \in L$ dan wel $x \notin L$. Evenzo betekent $L \in \mathcal{RE}$, dat er een effectieve methode bestaat om een (eventueel oneindige) lijst te maken, waar precies de elementen van L in staan. Voor formele definities van $\mathcal{R}$ en $\mathcal{RE}$ verwijzen we naar Toegepaste logica I.

De algemeenheid van de definitie van grammatica blijkt uit het feit dat $\mathcal{L}_0 = \mathcal{RE}$. We beginnen nu met de beperking S: alle regels hebben de gedaante (rAs, rqs), $A \in I_N$ en r,q,s alle in I^*. Triviaal is dat $\mathcal{L}_S \subset \mathcal{L}_0$; er geldt echter $\mathcal{L}_S = \mathcal{L}_0$. Deze beperking helpt dus nog niet veel.
Een geheel andere beperking is ND: alle regels hebben de gedaante (u,v) met $\ell(u) \leqslant \ell(v)$. Nu geldt $\mathcal{L}_{ND} \subset \mathcal{R}$. Dit levert wel een wezenlijke beperking. Bekend is namelijk $\mathcal{R} \subsetneqq \mathcal{RE}$, dus $\mathcal{L}_{ND} \subsetneqq \mathcal{L}_0$ (ND staat voor: non-decreasing). Leggen we de beperkingen S en ND beide op, dan ontstaat CS_0: alle regels hebben de gedaante (rAs, rqs), $A \in I_N$, $q \in I^+$, $r \in I^*$, $s \in I^*$. Triviaal is $\mathcal{L}_{CS_0} \subset \mathcal{L}_{ND}$; er geldt echter $\mathcal{L}_{CS_0} = \mathcal{L}_{ND}$. Verder is triviaal dat als $L \in \mathcal{L}_{ND}$, dan $\Lambda \notin L$. Dit is vervelend, omdat we ook talen wensen te beschouwen, die wel de lege symboolrij bevatten. Een kleine modificatie bewerkstelligt dit en leidt tot een belangrijke klasse. We beschouwen de beperking CS of type 1 (contekstgevoelig; Engels: context-sensitive): aan regels die alle van de gedaante (rAs, rqs), $A \in I_N$, $q \in I^+$, $r \in I^*$, $s \in I^*$ zijn, kan eventueel nog worden toegevoegd de regel (Z,Λ); als deze laatste regel voorkomt wordt ook nog geëist, dat in alle regels (u,v) het symbool Z niet in v voorkomt.

De bijbehorende klasse van talen heet $\mathcal{L}_1$ (talen van type 1 of <u>contekstgevoelige</u> talen). Triviaal is $\mathcal{L}_{CS_0} \subsetneqq \mathcal{L}_1$, ($\neq$ volgt uit $\{\Lambda\} \in \mathcal{L}_1$). Er geldt echter ook $\mathcal{L}_1 \subsetneqq \mathcal{R}$, dus ook $\mathcal{L}_1 \subsetneqq \mathcal{L}_0$.

De volgende beperking is CF of type 2 (contekstvrij; Engels: context-free): alle regels hebben de gedaante (A,q), $A \in I_N$, $q \in I^*$. De bijbehorende klasse van talen heet $\mathcal{L}_2$ (talen van type 2 of <u>contekstvrije</u> talen). Triviaal is $\mathcal{L}_2 \subset \mathcal{L}_0$. Er geldt echter ook $\mathcal{L}_2 \subsetneqq \mathcal{L}_1$.

De laatste beperking die we hier bespreken, is type 3 of <u>rechtslineair</u>: alle regels zijn van de gedaante (A,qB) of (A,q), waarin $A \in I_N$, $B \in I_N$, $q \in I_T^*$. De bijbehorende klasse van talen heet $\mathcal{L}_3$ (talen van type 3). Triviaal is $\mathcal{L}_3 \subset \mathcal{L}_2$; er geldt zelfs $\mathcal{L}_3 \subsetneqq \mathcal{L}_2$.

De hierboven gedane beweringen zullen in dit college niet allen worden bewezen. We ontlenen $\mathcal{R} \subsetneqq \mathcal{RE}$ aan Toegepaste logica I. De volgende tabel geeft aan wat in het vervolg wel en niet zal worden aangetoond.

bewijs van	geen bewijs van
$\mathcal{L}_{ND} \subset \mathcal{R}$	$\mathcal{L}_0 = \mathcal{RE}$
$\mathcal{L}_1 \subset \mathcal{R}$	$\mathcal{L}_S \supset \mathcal{L}_0$
$\mathcal{L}_2 \subsetneqq \mathcal{L}_1$	$\mathcal{L}_{CS_0} \supset \mathcal{L}_{ND}$
$\mathcal{L}_3 \neq \mathcal{L}_2$	$\mathcal{L}_1 \neq \mathcal{R}$

Voor de klassen die bij de typen van Chomsky horen (de zogenaamde <u>hiërarchie van Chomsky</u>) geldt:

$$\mathcal{L}_3 \subsetneqq \mathcal{L}_2 \subsetneqq \mathcal{L}_1 \subsetneqq \mathcal{L}_0 \quad .$$

HOOFDSTUK 1. REGULIERE TALEN

§2. EINDIGE AUTOMATEN

We bespreken nu een heel andere methode om talen te verkrijgen dan de in Hoofdstuk 0 besproken grammatica's.

Definitie eindige automaat

Een 5-tupel $A = \langle S, I, \delta, s_0, F \rangle$ heet eindige automaat als geldt:
S is een eindige verzameling (elementen van S heten toestanden),
I is een niet-lege eindige verzameling (elementen van I heten symbolen),
δ is een afbeelding $S \times I \to S$,
$s_0 \in S$ (begintoestand),
$F \subset S$ (elementen van F heten toegelaten toestanden).

De afbeelding δ wordt wel overgangsfunctie genoemd (Engels: transition function of next state function).

We breiden δ uit tot een afbeelding δ': $S \times I^* \to S$ en wel met recurrentie langs de opbouw van de elementen van I^*. Stel $x \in I^*$, $\xi \in I$, $s \in S$, dan

$$\delta'(s,\Lambda) := s,$$
$$\delta'(s,x\xi) := \delta(\delta'(s,x),\xi) .$$

Deze functie correspondeert met het beeld dat de automaat een doos is, die zich in een toestand s bevindt en bij invoer van een symbool ξ overgaat in toestand $\delta(s,\xi)$. Een symboolrij wordt symbool voor symbool van links naar rechts ingevoerd; δ' beschrijft het effect op de toestand bij invoer van de symboolrij.

Dat δ' inderdaad een voortzetting van δ is volgt uit

$$\delta'(s,\xi) = \delta'(s,\Lambda\xi) = \delta(\delta'(s,\Lambda),\xi) = \delta(s,\xi).$$

We laten voortaan het accent weg en schrijven voor δ' ook δ.

Definitie taal geaccepteerd door eindige automaat

Stel $A = \langle S,I,\delta,s_0,F\rangle$ een eindige automaat. De taal $L(A)$, geaccepteerd door A wordt gedefinieerd door

$$L(A) := \{x \in I^* | \delta(s_0,x) \in F\}.$$

Bij het vergelijken van talen die door een grammatica gegenereerd zijn met talen, die door een automaat geaccepteerd zijn, dient men te bedenken dat de symboolverzameling in het ene geval I_T en in het andere geval I heet.

Voor de functie δ geldt een concatenatieformule:

$$\text{als } x \in I^*,\ y \in I^*,\ s \in S, \text{ dan } \delta(s,xy) = \delta(\delta(s,x),y).$$

Dit bewijst men met recurrentie naar de opbouw van y:

$$\delta(s,x\Lambda) = \delta(s,x) = \delta(\delta(s,x),\Lambda)\ ,$$

$$\delta(s,xy\eta) = \delta(\delta(s,xy),\eta) = \delta(\delta(\delta(s,x),y),\eta) = \delta(\delta(s,x),y\eta)\ .$$

Definitie reguliere taal

Stel I een verzameling symbolen. Een deelverzameling L van I^* heet een reguliere taal als er een eindige automaat A met I als verzameling symbolen bestaat zo, dat $L = L(A)$.

§3. STELLING VAN MYHILL-NERODE

We maken in het vervolg gebruik van equivalentierelaties. Bij een equivalentierelatie r op een verzameling R hoort een partitie, waarvan de elementen de equivalentieklassen van r zijn. Deze partitie noteren we R/r; als $x \in R$, dan noteren we de equivalentieklasse die x bevat als $[x]_r$ of kortweg $[x]$. Het aantal equivalentieklassen noemen we de <u>index</u> van de equivalentierelatie: $\mathrm{index}(r) = \#R/r$.

Als R een semigroep is dan is een <u>congruentierelatie</u> $\equiv$ op R een equivalentierelatie op R waarvoor bovendien geldt:

als $x \in R$, $y \in R$, $z \in R$, $u \in R$, $x \equiv y$, $z \equiv u$, dan $xz \equiv yu$.

Een iets zwakker begrip is een <u>rechtsinvariante</u> (resp. <u>linksinvariante</u>) equivalentierelatie:

als $x \in R$, $y \in R$, $z \in R$, $x \equiv y$, dan $xz \equiv yz$ (resp. $zx \equiv zy$).

Een equivalentierelatie die zowel rechtsinvariant als linksinvariant is, is een congruentierelatie: als $x \equiv y$ en $z \equiv u$ dan $xz \equiv yz$ en $yz \equiv yu$, dus $xz \equiv yu$.

We passen dit nu toe op talen. Stel I een eindige verzameling en $L \subset I^*$. We definiëren een relatie e_L op I^* door voor $x \in I^*$ en $y \in I^*$ te stellen:

$$xe_Ly :\leftrightarrow \forall_{z \in I^*}[xz \in L \leftrightarrow yz \in L].$$

Het is triviaal dat e_L een rechtsinvariante equivalentierelatie is. Verder geldt: als $x \in L$ en xe_Ly, dan $y \in L$. Immers, uit $x \in L$ volgt $x\Lambda = x \in L$; uit xe_Ly volgt dan $y = y\Lambda \in L$.

Anders geformuleerd: als $x \in L$, dan $[x]_{e_L} \subset L$, dat wil zeggen dat L vereniging is van equivalentieklassen. Nog anders geformuleerd: een equivalentieklasse ligt geheel in L of heeft een lege doorsnede met L.

<u>Stelling 3.1</u> (<u>Myhill-Nerode</u>) Stel I eindige verzameling, $L \subset I^*$. Dan geldt:

$$L \text{ is regulier} \Leftrightarrow \text{index}(e_L) \text{ is eindig.}$$

<u>Bewijs</u>: Stel L regulier. Dan is er een eindige automaat $A = \langle S, I, \delta, s_0, F \rangle$ zo, dat $L = L(A)$. Stel $x \in I^*$, $y \in I^*$, dan geldt: als $\delta(s_0,x) = \delta(s_0,y)$, dan xe_Ly. Immers, als $z \in I^*$, $xz \in L$, dan $\delta(s_0,xz) \in F$, maar $\delta(s_0,xz) = \delta(\delta(s_0,x),z) = \delta(\delta(s_0,y),z) = \delta(s_0,yz)$, dus $\delta(s_0,yz) \in F$, dus $yz \in L$. Omgekeerd volgt op analoge wijze uit $yz \in L$, dat $xz \in L$. Dus xe_Ly. We definiëren nu $S' := \{s \in S \mid \exists_{x \in I^*}[\delta(s_0,x) = s]\}$ (de elementen van S' worden wel bereikbare toestanden genoemd).
De afbeelding $\varphi: S' \to I^*/e_L$, wordt gedefinieerd door

$$\varphi(s) := [x]_{e_L} \text{ , als } s \in S' \text{ en } \delta(s_0,x) = s.$$

Dit is een fatsoenlijke definitie. Omdat $s \in S'$, bestaat er namelijk een $x \in I^*$ zo, dat $\delta(s_0,x) = s$ en als $s = \delta(s_0,x) = \delta(s_0,y)$, dan xe_Ly en dus $[x]_{e_L} = [y]_{e_L}$. De afbeelding φ is echter ook surjectief want als $x \in I^*$, dan $\varphi(\delta(s_0,x)) = [x]_{e_L}$. Omdat S' eindig is, volgt daaruit dat I^*/e_L en dus $\text{index}(e_L)$ eindig is.
Stel nu omgekeerd dat index (e_L) eindig is. We definiëren een eindige automaat $A_0 = \langle S_0, I, \delta_0, s_{00}, F_0 \rangle$. We kiezen $S_0 := I^*/e_L$; dit is een eindige verzameling. Als $K \in S_0$ en $\xi \in I$, dan definiëren we $\delta_0(K,\xi)$ door een $x \in K$

te kiezen en $\delta_0(K,\xi) = [x\xi]_{e_L}$ te stellen. We moeten aantonen dat dit onafhankelijk is van de keuze van x in K. Stel $y \in K$, dan xe_Ly, dus $x\xi e_L y\xi$ (rechtsinvariant), dus $[x\xi]_{e_L} = [y\xi]_{e_L}$. We definiëren $s_{00} := [\Lambda]_{e_L}$ en $F_0 := \{K \in S_0 | K \cap L \neq \emptyset\}$. Dit levert dus een eindige automaat. We bewijzen nu eerst voor $x \in I^*$, dat $\delta_0(s_{00},x) = [x]_{e_L}$.
We doen dat door recurrentie naar de opbouw van x. Nu is $\delta_0(s_{00},\Lambda) =$ $= s_{00} = [\Lambda]_{e_L}$. Stel $\delta_0(s_{00},x) = [x]_{e_L}$, dan

$$\delta_0(s_{00},x\xi) = \delta_0(\delta_0(s_{00},x),\xi) = \delta_0([x]_{e_L},\xi) = [x\xi]_{e_L} .$$

Nu tonen we aan dat $L = L(A_0)$. Stel $x \in L$, dan $[x]_{e_L} \cap L \neq \emptyset$, dus $\delta_0(s_{00},x) =$ $= [x]_{e_L} \in F_0$, dus $x \in L(A_0)$. Stel nu $x \in L(A_0)$, dan $[x]_{e_L} = \delta(s_{00},x) \in F_0$, dus $[x]_{e_L} \cap L \neq \emptyset$, dus $[x]_{e_L} \subset L$, dus $x \in L$. Hieruit volgt dat L regulier is. ∎

Uit het bewijs van Stelling 3.1 valt nog wat meer te halen. Als L een reguliere taal is en A een eindige automaat, die L accepteert, dan levert het eerste deel van het bewijs, dat $\#S \geqslant \#S' \geqslant \#I^*/e_L = \text{index}(e_L)$. Het tweede deel van het bewijs levert het bestaan van een eindige automaat, die L accepteert, met een aantal toestanden = $\text{index}(e_L)$. Conclusie: $\text{index}(e_L)$ is het minimum van de aantallen toestanden van L accepterende eindige automaten. We kunnen echter ook nog afleiden dat er 'op isomorfie na' slechts één L accepterende eindige automaat bestaat met dat minimale aantal toestanden. Daartoe moeten we isomorfie van eindige automaten definiëren.

Definitie isomorfe eindige automaten

Stel $A_1 = \langle S_1,I,\delta_1,s_{10},F_1\rangle$ en $A_2 = \langle S_2,I,\delta_2,s_{20},F_2\rangle$ eindige automaten.

Dan heet A_1 isomorf A_2 als er een bijectie $\varphi\colon S_1 \to S_2$ bestaat zo, dat geldt:

$$\forall_{s \in S_1} \forall_{\xi \in I} [\varphi(\delta_1(s,\xi)) = \delta_2(\varphi(s),\xi)] \ , \ \varphi(s_{10}) = s_{20}$$

en $\{\varphi(s) | s \in F_1\} = F_2$.

Uiteraard is isomorfie een equivalentierelatie. Als nu L een reguliere taal is en A een eindige automaat waarvoor $L = L(A)$, en het aantal toestanden van A is index(e_L), dan volgt uit het eerste deel van het bewijs van Stelling 3.1 dat index(e_L) = #S ⩾ #S' ⩾ index(e_L). Daaruit volgt dat S' = S en dat de in dat bewijs gedefinieerde φ een bijectie $S \to I^*/e_L = S_0$ is. Deze brengt nu een isomorfie van A en A_0 tot stand. Stel namelijk $s \in S$ en $\xi \in I$. Omdat S' = S is er een $x \in I^*$ zo, dat $s = \delta(s_0,x)$ en $\varphi(\delta(s,\xi)) =$ $= \varphi(\delta(\delta(s_0,x),\xi)) = \varphi(\delta(s_0,x\xi)) = [x\xi]_{e_L} = \delta_0([x]_{e_L},\xi) = \delta_0(\varphi(s),\xi)$. Verder $\varphi(s_0) = \varphi(\delta(s_0,\Lambda)) = [\Lambda]_{e_L} = s_{00}$. Stel $s \in F$, $s = \delta(s_0,x)$, dan $x \in L$ en $\varphi(s) = [x]_{e_L} \in F_0$; omgekeerd stel $K \in F_0$, dan $K \cap L \neq \emptyset$, dus er is een $x \in K \cap L$ en $K = [x]_{e_L} = \varphi(\delta(s_0,x))$ en $\delta(s_0,x) \in F$. Daarmee is de isomorfie aangetoond. Dit levert de volgende stelling.

<u>Stelling 3.2</u>. Stel L een reguliere taal, A_1 en A_2 eindige automaten zo, dat $L = L(A_1) = L(A_2)$ en van A_1 en A_2 is het aantal toestanden = index(e_L). Dan zijn A_1 en A_2 isomorf.

<u>Opmerking</u>. Myhill werkte in plaats van met de relatie e_L met f_L gedefinieerd door

$$xf_Ly :\Leftrightarrow \forall_{z \in I^*} \forall_{w \in I^*} [zxw \in L \Leftrightarrow zyw \in L].$$

Algebraïsch is deze relatie mooier dan e_L, omdat het een congruentierelatie is. Verder geldt een analoge stelling, omdat index(f_L) eindig $\Leftrightarrow$ index(e_L) eindig.

Het mooie verband tussen index (e_L) en het aantal toestanden van een accepterende eindige automaat heeft geen tegenhanger bij f_L.

We geven nu enkele voorbeelden van talen, waarvan we de regulariteit met behulp van de stelling van Myhill-Nerode onderzoeken. We merken eerst op dat als L een taal is, de verzameling $\{x \in I^* \mid \forall_{w\in I^*} [xw \notin L]\}$ een equivalentieklasse bij e_L is. Kies nu $a \in I^*$, $\ell(a) = \ell$ en $L = \{a\}$. De $\ell + 1$ beginstukken van a vormen elk een klasse met één element; alle andere woorden vormen ook een klasse. Dus index(e_L) $= \ell + 2$ en L is regulier. Om een automaat voor L te maken, schrijven we a in letters uit: $a = \xi_0 \ldots \xi_{\ell-1}$ en kiezen $S := \{0,\ldots,\ell+1\}$,

$$\delta(j,\xi) = \left\{ \begin{array}{l} j+1 \text{ als } \xi = \xi_j \\ \ell+1 \text{ als } \xi \neq \xi_j \end{array} \right\} \text{ voor } j \in \{0,\ldots,\ell-1\}\ ,\ \delta(\ell,\xi) = \delta(\ell+1,\xi) = \ell+1$$

voor alle $\xi \in I$. Tenslotte $s_0 = 0$, $F = \{\ell\}$.

Een speciaal geval is $a = \Lambda$, $\ell = 0$, $L = \{\Lambda\}$. Dan is $S = \{0,1\}$, $\delta(j,\xi) = 1$ voor $j \in S$, $\xi \in I$, $s_0 = 0$, $F = \{0\}$.

Voor een eindige taal L geldt index(e_L) $\leqslant 1 + \sum_{a\in L} (\ell(a) + 1)$; alle eindige talen zijn regulier. Een speoiaal geval is $L = \emptyset$, met index $e_L = 1$ en automaat met $S = \{0\}$, $\delta(0,\xi) = 0$ voor $\xi \in I$, $F = \emptyset$.

Stel nu $I = \{\alpha,\beta\}$, $L = \{\alpha^m\beta^n \mid m \in \mathbb{N}, n \in \mathbb{N}\}$. Deze taal heeft de eigenschap dat, als $x \in L$, ook ieder beginstuk van x in L zit. Daaruit volgt:
$\forall_{w\in I^*} [xw \notin L] \Leftrightarrow x \notin L$. Het complement van L is dus een equivalentieklasse.
Als $x = \alpha^m$, dan $\forall_{w\in I^*} [xw \in L \Leftrightarrow w \in L]$. Als $x = \alpha^m\beta^n$ met $n \geqslant 1$, dan $\forall_{w\in I^*} [xw \in L \Leftrightarrow \exists_{k\in\mathbb{N}} [w = \beta^k]]$.
Dus index(e_L) = 3; L is regulier. Voor een automaat kiezen we $S = \{0,1,2\}$.

$$\left.\begin{array}{l}\delta(0,\alpha) = 0\\ \delta(0,\beta) = 1\end{array}\right\} \left.\begin{array}{l}\delta(1,\alpha) = 2\\ \delta(1,\beta) = 1\end{array}\right\} \left.\begin{array}{l}\delta(2,\alpha) = 2\\ \delta(2,\beta) = 2\end{array}\right\},\ s_0 = 0,\ F = \{0,1\}.$$

Stel nu $I = \{\alpha,\beta\}$, $L = \{\alpha^n\beta^n \mid n \in \mathbb{N}\}$. Nu geldt voor $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $k \neq \ell$, dat $\alpha^k e_L \alpha^\ell$ niet geldt, immers $\alpha^k\beta^k \in L$, $\alpha^\ell\beta^k \notin L$. Hieruit volgt $\mathrm{index}(e_L) = \infty$, L is niet regulier. Op analoge wijze toont men aan, dat $\{\alpha^n\beta\alpha^n \mid n \in \mathbb{N}\}$ niet regulier is. We zullen later aantonen (Stelling 5.2), dat een taal regulier is dan en slechts dan als hij van type 3 is. De zojuist behandelde niet-reguliere talen zijn echter wel talen van type 2 op grond van de volgende grammatica's: $I_T = \{\alpha,\beta\}$, $I_N = \{Z\}$, $R = \{(Z,\alpha Z\beta),(Z,\Lambda)\}$, resp. $R = \{(Z,\alpha Z\alpha),(Z,\beta)\}$. Het bovenstaande levert dus een bijdrage tot de bewering $\mathcal{L}_3 \neq \mathcal{L}_2$.

§4. OPERATIES OP TALEN. REGULIERE EXPRESSIES

Als de symboolverzameling I gegeven is, zijn talen deelverzamelingen van I^*. Op talen kunnen de gebruikelijke verzamelingstheoretische operaties worden toegepast zoals vereniging, doorsnede en complement. We kunnen echter ook <u>concatenatie</u> van talen definiëren als volgt: als $L \subset I^*$, $M \subset I^*$, dan

$$LM := \{xy \mid x \in L,\ y \in M\}\ .$$

Deze concatenatie definieert dus een binaire operatie op $P(I^*)$, die associatief is. Er is verder een neutraal element, namelijk $\{\Lambda\}$: $\{\Lambda\}L = L\{\Lambda\} = L$ voor alle $L \in P(I^*)$. Op grond hiervan is $P(I^*)$ een monoïde. Naast het neutrale element is er ook een nulelement, namelijk $\emptyset$: $\emptyset L = L\emptyset = \emptyset$ voor alle $L \in P(I^*)$. Zoals in iedere monoïde kunnen we nu machten van talen definiëren, L^n voor alle $n \in \mathbb{N}$ door $L^0 := \{\Lambda\}$, $L^{n+1} = L^nL$. Inderdaad is dan $L^1 = L$, $L^2 = LL$.

Nog een operatie op talen is de overgang van L naar L^*; stel $L \in P(I^*)$, $L^* := \{x_0 \ldots x_{n-1} \mid n \in \mathbb{N}, \forall_{j \in \{0,\ldots,n-1\}} [x_j \in L]\}$. Het gebruik van de ster moet gerechtvaardigd worden; we hadden namelijk al eens een ster gebruikt voor de overgang van een symboolverzameling I naar de verzameling van alle symboolrijen I^*. Nu is $I \subset I^*$, zodat bovenstaande definitie ook op I toepasselijk is; het is echter duidelijk dat het resultaat dan de verzameling van alle symboolrijen is en beide opvattingen van de ster tot hetzelfde resultaat voeren en verwarring dus niet te vrezen is.

Het is duidelijk dat $L^* = \bigcup_{n=0}^{\infty} L^n$; we definiëren $L^+ = \bigcup_{n=1}^{\infty} L^n$. Er geldt $\Lambda \in L^*$ en $L^* = L^+ \cup \{\Lambda\}$. Het is echter mogelijk, dat $\Lambda \in L$ en dan geldt $L^* = L^+$. Algemener geldt

$$L^* = L^+ \Leftrightarrow \Lambda \in L \Leftrightarrow \forall_{j \in \mathbb{N}} [L^j \subset L^{j+1}] .$$

$$\emptyset^* = \{\Lambda\}, \; I^n = \{x \in I^* \mid \ell(x) = n\} .$$

Dit alles tonen we niet aan. De steroperatie wordt ook wel Kleene-ster genoemd.

Stelling 4.1. Stel I symboolverzameling, $L \subset I^*$, $M \subset I^*$. Dan geldt

als L en M regulier, dan $L \cup M$, $L \cap M$ en LM regulier;

als L regulier, dan $I^* \setminus L$ en L^* regulier.

Bewijs: Stel de eindige automaten $A_L = \langle S_L, I, \delta_L, s_{L0}, F_L \rangle$ zo, dat $L = L(A_L)$ en $A_M = \langle S_M, I, \delta_M, s_{M0}, F_M \rangle$ zo, dat $M = L(A_M)$. We definiëren nu $A_1 = \langle S_1, I, \delta_1, s_{10}, F_1 \rangle$ door $S_1 := S_L \times S_M$, $\delta_1((s,t),\xi) := (\delta_L(s,\xi), \delta_M(t,\xi))$ voor $s \in S_L$, $t \in S_M$, $\xi \in I$; $s_{10} = (s_{L0}, s_{M0})$; $F_1 := \{(s,t) \mid s \in F_L \text{ of } t \in F_M\}$. Met recurrentie naar de opbouw van $x \in I^*$ bewijst men dat

$$\delta_1(s_{10},x) = (\delta_L(s_{L0},x), \delta_M(s_{M0},x)) ,$$

en daaruit volgt $\delta_1(s_{10},x) \in F \Leftrightarrow \delta_L(s_{L0},x) \in F_L$ of $\delta_M(s_{M0},x) \in F_M \Leftrightarrow x \in L$ of $x \in M \Leftrightarrow x \in L \cup M$, dus $L(A_1) = L \cup M$.

Een automaat A_2 voor $L \cap M$, krijgt men door in A_1 de F_1 te vervangen door $F_2 := \{(s,t) \mid s \in F_L$ en $t \in F_M\}$. Een automaat A_3 voor $I^*\backslash L$ krijgt men door in A_L de F_L te vervangen door $F_3 := S_L\backslash F_L$. We definiëren nu een eindige automaat A_4 door $S_4 := S_L \times P(S_M)$. Voor $s \in S_L$, $V \subset S_M$, $\xi \in I$ stellen we

$$\delta_4((s,V),\xi) := (\delta_L(s,\xi),\{\delta_M(t,\xi) \mid t \in V\} \cup c),$$

waarin

$$c := \begin{cases} \{s_{M0}\}, & \text{als } \delta_L(s,\xi) \in F_L , \\ \emptyset \quad , & \text{als } \delta_L(s,\xi) \notin F_L . \end{cases}$$

$$s_{40} := \begin{cases} (s_{L0},\{s_{M0}\}), & \text{als } s_{L0} \in F_L , \\ (s_{L0}, \emptyset) \quad , & \text{als } s_{L0} \notin F_L . \end{cases}$$

$$F_4 := \{(s,V) \in S_L \times P(S_M) \mid V \cap F_M \neq \emptyset\} .$$

Met recurrentie naar de opbouw van $x \in I^*$ bewijst men dat $\delta_4(s_{40},x) = $ $=(\delta_L(s_{L0},x),\{\delta_M(s_{M0},w) \mid w \in I^* \wedge \exists_{y \in L} [yw = x]\})$. Hieruit volgt $\delta_4(s_{40},x) \in F_4 \Leftrightarrow \exists_{w \in I^*} \exists_{y \in L} [yw = x \wedge \delta_M(s_{M0},w) \in F_M] \Leftrightarrow \exists_{w \in M} \exists_{y \in L} [yw = x]$ $\Leftrightarrow x \in LM$, dus $L(A_4) = LM$. We definiëren nu een eindige automaat A_5 door $S_5 := P(S_L)$. Voor $V \subset S_L$ en $\xi \in I$ stellen we $\delta_5(V,\xi) := b \cup c$, waarin

$b := \{\delta_L(s,\xi) \mid s \in V\}$,

$$c := \begin{cases} \{s_{L0}\} & \text{als } b \cap F_L \neq \emptyset, \\ \emptyset & \text{als } b \cap F_L = \emptyset. \end{cases}$$

$s_{50} := \{s_{L0}\}$, $F_5 = \{V \in P(S_L) \mid V \cap F_L \neq \emptyset\}$.

Met recurrentie naar de opbouw van $x \in I^*$ bewijst men, dat $\delta_5(s_{50},x) = \{\delta_L(s_{L0},w) \mid w \in I^* \wedge \exists_{u \in L^*} [uw = x]\}$, en daaruit volgt

$\delta_5(s_{50},x) \in F_5 \Leftrightarrow \exists_{w\in I^*} \exists_{u\in L^*} [uw = x \wedge \delta_L(s_{L0},w) \in F_L] \Leftrightarrow \exists_{w\in L} \exists_{u\in L^*}[uw = x]$
$\Leftrightarrow x \in L^*L \Leftrightarrow x \in L^+$, dus $L(A_5) = L^+$, dus L^+ regulier. Omdat $L^* = L^+ \cup \{\Lambda\}$, is ook L^* regulier. ∎

We zullen aantonen dat alle reguliere talen te verkrijgen zijn uitgaande van eindige talen door herhaaldelijke toepassing van vereniging, concatenatie en steroperatie. Daarbij zijn echter niet alle eindige talen nodig, omdat sommige eindige talen uit andere eindige talen verkrijgbaar zijn met behulp van genoemde operaties. Allereerst is een niet-lege eindige verzameling vereniging van singletons (verzamelingen met één element):
$\{a_0,\ldots,a_{n-1}\}= \{a_0\} \cup \ldots \cup \{a_{n-1}\}$.Verder is een singleton van een niet-leeg woord concatenatie van singletons van éénletterwoorden:
$\{\xi_0\ldots\xi_{\ell-1}\} = \{\xi_0\} \ldots \{\xi_{\ell-1}\}$. Bedenkt men ook nog dat $\{\Lambda\} = \emptyset^*$, dan ziet men dat we kunnen volstaan met $\emptyset$ en $\{\xi\}$ voor alle $\xi \in I$.

Definitie Rec

Rec is de kleinste onder de verzamelingen F van talen, die voldoen aan:

$\emptyset \in F$

$\{\xi\} \in F$ voor alle $\xi \in I$,

als $L \in F$ en $M \in F$, dan $L \cup M \in F$ en $LM \in F$,

als $L \in F$, dan $L^* \in F$.

Met het woord kleinste is bedoeld dat voor alle F, die aan bovenstaande eisen voldoen, geldt Rec $\subset F$ en dat Rec zelf ook aan die eisen voldoet.

We tonen eerst aan dat Rec bestaat. Dit kan op twee manieren.

1. Er bestaan verzamelingen van talen, die aan de eisen voor F voldoen, bijvoorbeeld $P(I^*)$, de verzameling van alle talen. Verder is het zo, dat de doorsnede van een collectie van F's, die alle aan de eisen vol-

doen, ook aan de eisen voldoet. We vinden nu Rec als de doorsnede van alle F's die aan de eisen voldoen.

2. Neem de verzameling van alle talen, die ontstaan door een eindig, maar onbegrensd aantal malen operaties $(L,M) \mapsto L \cup M$, $(L,M) \mapsto LM$ en $L \mapsto L^*$ toe te passen, uitgaande van de talen $\emptyset$ en $\{\xi\}$ voor $\xi \in I$. Dat levert een verzameling talen, die aan de eisen voldoet en kennelijk ook de kleinste.

Stelling 4.2. Rec = verzameling der reguliere talen.

Bewijs: Op grond van de resultaten van §3 en Stelling 4.1 is de verzameling der reguliere talen een verzameling die aan de eisen voor F in de definitie van Rec voldoet, dus Rec $\subset$ verzameling der reguliere talen. Stel nu L een reguliere taal en $A = \langle S,I,\delta,s_0,F\rangle$ een eindige automaat zo, dat $L = L(A)$. We nummeren de elementen van S: stel $\#S = n$ en $S = \{s_0,\dots,s_{n-1}\}$; ook hier is s_0 de begintoestand. We definiëren nu talen $L(i,j,k)$ voor $\{i,j\}\subset\{0,\dots,n-1\}$ en $k \in \{0,\dots,n\}$, als volgt

$$L(i,j,k) := \{x \in I^* \mid \delta(s_i,x) = s_j \wedge \forall_{y\in I^+}\forall_{z\in I^+}[yz = x \wedge \delta(s_i,y) = s_\ell \Rightarrow \ell < k]\} .$$

We bewijzen dat alle $L(i,j,k) \in$ Rec en wel met volledige inductie naar k. Als $x \in L(i,j,0)$, dan $\ell(x) \leq 1$, dus $L(i,j,0)$ eindig, dus $L(i,j,0) \in$ Rec. Stel nu de bewering juist voor zekere $k < n$. Er geldt

$$L(i,j,k+1) = L(i,j,k) \cup L(i,k,k)(L(k,k,k))^*L(k,j,k).$$

Dat ieder element van het rechterlid ook element is van het linkerlid, is duidelijk. Een element x van het linkerlid voeren we door de automaat A

beginnend in toestand s_i. Het eindigt dan in toestand s_j. Als alle toestanden nummers $< k$ hebben dan $x \in L(i,j,k)$. Zijn er tussentoestanden met nummer k, dan verknippen we het woord tot een concatenatie van woorden bij iedere plaats waar een toestand met nummer k optreedt. Het voorste woord in deze concatenatie zit in $L(i,k,k)$, het achterste in $L(k,j,k)$ en eventuele tussengelegen woorden in $L(k,k,k)$. Dus $x \in$ rechterlid. Volgens inductieveronderstelling zijn alle talen in het rechterlid elementen van $\mathcal{R}ec$, dus ook $L(i,j,k+1) \in \mathcal{R}ec$. Nu is $L(i,j,n) = \{x \in I^* | \delta(s_i,x) = s_j\}$ en dus $L = \bigcup_{s_j \in F} L(0,j,n)$. Daaruit volgt $L \in \mathcal{R}ec$. ∎

In Stelling 4.2 wordt een zogenaamde recurrente karakterisering van de reguliere talen tot stand gebracht. Op grond daarvan kan men van een eigenschap van talen aantonen dat zij voor alle reguliere talen geldt, door aan te tonen dat $\emptyset$ en $\{\xi\}$ voor alle $\xi \in I$ de eigenschap hebben benevens dat als L en M de eigenschap hebben, dit ook geldt voor $L \cup M$, LM en L^*. Een voorbeeld hiervan levert de gespiegelde L^T van een taal L, gedefinieerd door $L^T := \{x^T | x \in L\}$.

<u>Stelling 4.3</u>. Als de taal L regulier is, dan is L^T regulier.
Men kan dit bewijzen door het bovenstaande toe te passen op de eigenschap "L^T is regulier".

<u>Opgave</u>. Voer dit uit.

Als men een reguliere taal opbouwt met een recurrent proces, dan kan men dit proces weergeven met een expressie, die uitdrukt welke operaties successievelijk zijn toegepast. Zo is $\{\alpha^m\beta^n | m \in \mathbb{N}, n \in \mathbb{N}\} = \{\alpha\}^*\{\beta\}^*$. We laten nu de accolades weg en komen zo tot reguliere expressies.

<u>Definitie reguliere expressie</u>

Stel I een verzameling symbolen, $I \cap \{(,),\emptyset,\cup,*\} = \emptyset$.

Als $\alpha \in I$, dan is α een reguliere expressie;

$\emptyset$ is een reguliere expressie;

als E en F reguliere expressies zijn, dan zijn $(E \cup F)$ en (EF) reguliere expressies;

als E een reguliere expressie is, dan is $E*$ een reguliere expressie;

alleen hetgeen op grond van bovenstaande voorschriften is gevormd, is een reguliere expressie.

Voorbeelden van reguliere expressies zijn $\emptyset$, $\emptyset*$, $(\alpha \cup \beta)$, $((\alpha \cup \beta)(\beta \cup \gamma))*$.

We definiëren nu de taal L(E) van een reguliere expressie E door recurrentie langs de opbouw van E.

<u>Definitie taal van een reguliere expressie</u>

$L(\emptyset) := \emptyset$;

als $\alpha \in I$, dan $L(\alpha) := \{\alpha\}$;

als E en F reguliere expressies zijn, dan $L((E \cup F)) := L(E) \cup L(F)$ en $L((EF)) := L(E)L(F)$;

als E een reguliere expressie is, dan $L(E*) = (L(E))^*$.

<u>Stelling 4.4</u>, (<u>Kleene</u>). Stel I symboolverzameling, $L \subset I^*$. Dan geldt:

L is regulier $\Leftrightarrow$ er bestaat een reguliere expressie zo, dat $L = L(E)$.

Dit volgt direct uit Stelling 4.2.

Bij verschillende reguliere expressies kan dezelfde taal behoren; noem deze equivalent: $E \equiv F :\Leftrightarrow L(E) = L(F)$.

Enkele voorbeelden van equivalente reguliere expressies:

$$(\alpha \cup \beta)* \equiv (\alpha * \beta *)* ,$$

$$(\alpha \cup \beta)* \equiv ((\alpha * \beta *)* \alpha *) ,$$

$$(\alpha * \beta)* \equiv (\emptyset * \cup ((\alpha \cup \beta)* \beta)) .$$

§5. TALEN VAN TYPE 3

Een grammatica is van type 3 als alle regels van de gedaante (A,qB) of (A,q) zijn met $A \in I_N$, $B \in I_N$, $q \in I_T^*$. Voor afleidingen bij een grammatica van type 3 geldt het volgende:

Als $p \overset{*}{\Rightarrow} r$, dan is het aantal voorkomens van hulpsymbolen in $r \leqslant$ het aantal voorkomens van hulpsymbolen in p.

Als $Z \overset{*}{\Rightarrow} r$, dan $r \in I_T^*$ of $r = sB$ met $s \in I_T^*$, $B \in I_N$.

Als $Z \overset{*}{\Rightarrow} r$, $r \in I_T^*$, dan bestaat een afleiding uit een aantal malen toepassing van een regel van de gedaante (A,qB) (dit aantal kan nul zijn), gevolgd door één toepassing van een regel van de gedaante (A,q).

We willen nu aantonen dat we ons kunnen beperken tot het geval dat voor alle regels van de gedaante (A,qB) geldt $\ell(q) = 1$ en voor alle regels van de gedaante (A,q) geldt $\ell(q) = 0$, zonder dat daardoor de klasse der gegenereerde talen wordt verkleind. We zullen zulke regels tijdelijk "toegelaten" regels noemen: $(A,\alpha B)$ of (A,Λ) met $A \in I_N$, $B \in I_N$, $\alpha \in I_T$. De overige regels noemen we "verboden": (A,qB) met $\ell(q) \geqslant 2$, (A,q) met $\ell(q) \geqslant 1$, (A,B).

<u>Stelling 5.1</u>. Bij iedere grammatica G van type 3 bestaat een grammatica G' van type 3 zo, dat $L(G') = L(G)$ en alle regels van G' de gedaante $(A,\alpha B)$ of (A,Λ) hebben met $A \in I_N$, $B \in I_N$, $\alpha \in I_T$.

Bewijs: We onderscheiden twee gevallen.

1. Alle verboden regels van G zijn van de gedaante (A,B). We passen dan volledige inductie toe naar het aantal hulpsymbolen, die in rechterleden van verboden regels voorkomen. Als dit aantal nul is, dan zijn er geen verboden regels en kiezen we $G' = G$. Als het aantal >0 is kiezen we een hulpsymbool C, dat in een rechterlid van een verboden regel voorkomt. We construeren een grammatica G'', die alleen in de verzameling regels van G verschilt. Deze verzameling regels krijgen we door wijziging van de verzameling regels van G en wel door voor iedere regel van de gedaante (P,C) in G deze regel weg te laten en voor iedere regel van de gedaante (C,r) met $r \neq C$ in G een overeenkomstige regel (P,r) toe te voegen. Het is duidelijk dat ook in G'' alle verboden regels van de gedaante (A,B) zijn en dat het aantal hulpsymbolen, die in rechterleden van verboden regels voorkomen één minder is dan bij G. Op grond van de inductieveronderstelling is er een grammatica G' met uitsluitend toegelaten regels zo, dat $L(G') = L(G'')$, We tonen aan dat $L(G'') = L(G)$. Stel eerst $x \in L(G'')$ en beschouw een afleiding van x behorend bij G''. Als bij een stap van die afleiding een regel is toegepast, die geen regel van G is, dan is dat een regel van de gedaante (P,r) als hierboven ingevoerd en de afleidingsstap is uP, ur. Bij G kan dit door twee afleidingsstappen worden vervangen: uP, uC, ur. Zo kan de afleiding omgezet worden in een afleiding bij G, zodat $x \in L(G)$. Gaan we omgekeerd uit van $x \in L(G)$ met een afleiding bij G, dan mogen we aannemen dat bij de afleiding geen regels van de gedaante (A,A) zijn toegepast, omdat zo'n toepassing de symboolrij niet verandert en de bijbehorende stap dus overgeslagen kan worden. Een toepassing van een regel die niet in G'' voorkomt, gebruikt een regel (P,C) en de afleidingsstap is uP, uC. De volgende afleidingsstap levert dan ur, met toepassing van een regel (C,r) met $r \neq C$.

In G" kunnen deze twee stappen tot één samengetrokken worden: uP, ur met toepassing van de regel (P,r). Zo wordt de afleiding omgezet in een afleiding bij G", zodat $x \in L(G")$. Daarmee is $L(G") = L(G)$ aangetoond en dus ook $L(G') = L(G)$.

2. In G komt een verboden regel voor die niet van de gedaante (A,B) is. We passen nu volledige inductie toe naar het aantal verboden regels. We beschouwen eerst het geval dat er een regel is van de gedaante (A,qB) met $\ell(q) \geq 2$. Kies er zo een: $(A,\xi_0 \ldots \xi_{\ell-1}B)$ met $\ell \geq 2$, $\xi_0,\ldots,\xi_{\ell-1}$ alle in I_T. We definiëren nu een grammatica G". De verzameling hulpsymbolen bestaat uit de hulpsymbolen van G met daaraan toegevoegd $\ell-1$ nieuwe hulpsymbolen $X_0,\ldots,X_{\ell-2}$. De verzameling regels ontstaat uit de verzameling regels van G door de regel $(A,\xi_0 \ldots \xi_{\ell-1}B)$ weg te laten en de volgende regels toe te voegen:

$$(A,\xi_0X_0) \quad ,$$
$$(X_j,\xi_{j+1}X_{j+1}) \quad \text{voor } j \in \{0,\ldots,\ell-3\},$$
$$(X_{\ell-2},\xi_{\ell-1}B) \ .$$

Het is duidelijk dat in G" het aantal verboden regels één minder is dan in G, zodat op grond van de inductievenonderstelling of op grond van geval 1 een grammatica G' met uitsluitend toegelaten regels bestaat zo, dat $L(G') = L(G")$. We tonen aan dat $L(G") = L(G)$. Stel eerst $x \in L(G)$ en beschouw een afleiding van x bij G. Een toepassing van de regel $(A,\xi_0 \ldots \xi_{\ell-1}B)$ kan bij G" door ℓ stappen vervangen worden die successievelijk de bij deze regel behorende ℓ regels gebruiken. Daaruit volgt dat $x \in L(G")$. Omgekeerd stel $x \in L(G")$ met een afleiding bij G". Toepassing van regels van G kan nooit één van de nieuwe hulpsymbolen opleveren. Als in de afleiding regels worden toegepast die niet in G voorkomen, dan moet dit de eerste maal dat dit gebeurt de regel (A,ξ_0X_0) zijn. Er is slechts één regel die X_0 in het linkerlid heeft enz. Zo voortgaande ziet men, dat het bovenstaande rijtje regels in volgorde achter elkaar moet worden toegepast. In G kunnen deze ℓ stappen vervangen worden door de ene stap met toepassing van $(A,\xi_0\ldots\xi_{\ell-1}B)$. Door herhaalde toepassing vinden we een afleiding bij G en dus $x \in L(G)$. Daarmee is $L(G'') = L(G)$ aangetoond en dus ook $L(G') = L(G)$.

Het geval dat we een verboden regel hebben van de gedaante $(A,\xi_0\ldots\xi_{\ell-1})$ met $\ell \geqslant 1$ wordt op analoge wijze behandeld met ℓ nieuwe hulpsymbolen $X_0,\ldots,X_{\ell-1}$ en ter vervanging van de regel $(A,\xi_0\ldots\xi_{\ell-1})$ de regels

$$(A,\xi_0X_0)\ ,$$
$$(X_j,\xi_{j+1}X_{j+1}) \text{ voor } j \in \{0,\ldots,\ell-2\}\ ,$$
$$(X_{\ell-1},\Lambda)\ .$$

We werken dit niet verder uit. ∎

Als een grammatica G van type 3 uitsluitend toegelaten regels heeft, dan bestaat een afleiding van een woord $x \in L(G)$ uit $\ell(x)+1$ stappen. Na k stappen $(k \leqslant \ell(x))$ is een woord verkregen van de gedaante uA met $u \in I_T^*$, $\ell(u) = k$, $A \in I_N$ en u is een beginstuk van x.

<u>Stelling 5.2.</u> L_3 = verzameling der reguliere talen.

<u>Bewijs</u>: Stel $L \in L_3$. Op grond van Stelling 5.1 bestaat er dan een grammatica G van type 3 zo, dat $L = L(G)$ en alle regels van G zijn van de gedaante $(A,\alpha B)$ of (A,Λ) met $A \in I_N$, $B \in I_N$, $\alpha \in I_T$. We definiëren een eindige automaat $A = \langle S,I_T,\delta,s_0,F\rangle$ door $S := P(I_N)$, $\delta(V,\xi) := \{B \in I_N \mid \exists_{C\in V}[(C,\xi B) \in R]\}$ voor $V \in S$, $\xi \in I_T$, $s_0 := \{Z\}$, $F := \{V \in S \mid \exists_{C\in V}[(C,\Lambda) \in R]\}$. Met recurrentie

langs de opbouw van $x \in I_T^*$ bewijst men dat $\delta(s_0,x) = \{B \in I_N | Z \overset{*}{\underset{G}{\Rightarrow}} xB\}$ en daaruit volgt $\delta(s_0,x) \in F \Leftrightarrow \exists_{B \in I_N} [Z \overset{*}{\underset{G}{\Rightarrow}} xB \wedge (B,\Lambda) \in R] \Leftrightarrow Z \overset{*}{\underset{G}{\Rightarrow}} x$, dus $L(A) = L(G) = L$.

Stel nu omgekeerd L regulier. Er bestaat dan een eindige automaat $A = \langle S,I,\delta, s_0,F\rangle$ zo, dat $L = L(A)$. We mogen veronderstellen dat $S \cap I^* = \emptyset$; anders vervangen we S door een andere verzameling met aanpassing van δ, s_0 en F. We definiëren nu een grammatica G als volgt: $I_N := S$, $I_T := I$, $Z := s_0$, $R := \{(s,\xi\delta(s,\xi)) | s \in S, \xi \in I_T\} \cup \{(s,\Lambda) | s \in F\}$. Het is duidelijk dat G een grammatica van type 3 is. Verder geldt voor $s \in S$, $x \in I_T^*$ dat $(\delta(s,x),\xi\delta(s,x\xi)) \in R$. Stel nu $x = \xi_0 \ldots \xi_{\ell-1} \in L(A)$, dan levert $X_j :=$ $:= \xi_0 \ldots \xi_{j-1}\delta(s_0,\xi_0 \ldots \xi_{j-1})$ voor $j \in \{0,\ldots,\ell\}$ en $X_{\ell+1} := \xi_0 \ldots \xi_{\ell-1}$ een afleiding $X_0,\ldots,X_{\ell+1}$ van x bij G. Stel nu omgekeerd $X_0,\ldots,X_{\ell+1}$ een afleiding bij G van $\xi_0 \ldots \xi_{\ell-1}$, dan geldt $X_j = \xi_0 \ldots \xi_{j-1}\delta(s_0,\xi_0 \ldots \xi_{j-1})$ voor $j \in \{0,\ldots,\ell\}$ en $\delta(s_0,\xi_0 \ldots \xi_{\ell-1}) \in F$. ∎

Samenvattend kunnen we stellen dat de Stellingen 3.1, 4.2, 4.4 en 5.2 ons in totaal vijf verschillende karakteriseringen van reguliere talen hebben opgeleverd (inclusief de als definitie gekozen karakterisering). Het voordeel is, dat we in elk voorkomend geval kunnen kiezen welke karakterisering ons het beste uitkomt. Dat het complement van een reguliere taal regulier is volgt gemakkelijk uit de definitie of uit Stelling 3.1 ($\text{index}(e_L) = \text{index}(e_{I^* \setminus L})$). Dat de gespiegelde van een reguliere taal regulier is volgt gemakkelijk uit Stelling 4.2 of 4.4 (zie ook Stelling 4.3). Dat een reguliere taal contekstvrij is volgt direct uit Stelling 5.2.

Men kan ook een grammatica spiegelen: een grammatica G veranderen in een grammatica G' door iedere regel (p,q) te vervangen door (p^T,q^T). Het is duidelijk dat $L(G') = (L(G))^T$. Doet men dit met een grammatica van type 3 dan ontstaat een grammatica, waarvan de regels van de gedaante (A,Bq) of (A,q) met $A \in I_N$, $B \in I_N$, $q \in I_T^*$ zijn; een dergelijke grammatica wordt <u>linkslineair</u> genoemd. Omdat grammatica's van type 3 reguliere talen opleveren en gespiegelden van reguliere talen weer regulier zijn leveren linkslineaire grammatica's ook reguliere talen op. Stopt men echter rechtslineaire en linkslineaire regels in één grammatica, dan hoeft dit niet meer het geval te zijn. Zo genereert de grammatica met $I_N = \{Z,A\}, I_T = \{\alpha,\beta\}$, $R = \{(Z,\alpha A),(A,Z\beta),(Z,\Lambda)\}$ de taal $\{\alpha^n\beta^n | n \in \mathbb{N}\}$, die niet regulier is.

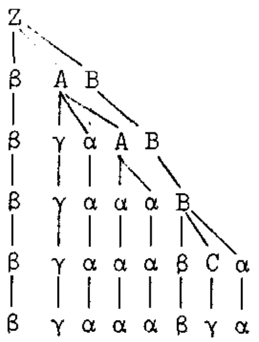
Het is duidelijk dat de gespiegelde van een contekstvrije grammatica contekstvrij is. Dit levert de volgende stelling.

<u>Stelling 5.3</u>. Als L een contekstvrije taal is, dan is L^T een contekstvrije taal.

HOOFDSTUK 2. CONTEKSTVRIJE TALEN

§6. INLEIDING

In §1 is vastgesteld wat een contekstvrije grammatica is: alle regels zijn van de gedaante (A,q) met $A \in I_N$, $q \in I^*$. Bij een afleidingsstap wordt een hulpsymbool A vervangen door q, ongeacht de rest van de symboolrij waarin deze A voorkomt; dit verklaart de term "contekstvrij". Dit contekstvrije karakter maakt het mogelijk om een afleiding op een andere wijze te beschouwen. We leiden dat in met een voorbeeld van een afleiding, waarin we de optredende symboolrijen onder elkaar schrijven en met streepjes aangeven, hoe de regels worden toegepast.



Toegepaste regel:

(Z,βAB)

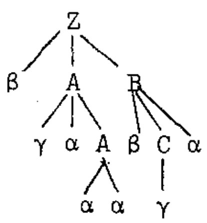
(A,γαA)

(A,αα)

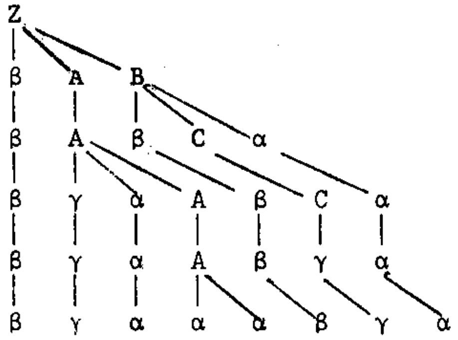
(B,βCα)

(C,γ)

Als men deze "boom" bekijkt, krijgt men de neiging de overgangen waarbij niets verandert weg te laten en kortweg de volgende figuur te tekenen:



Aan deze boom is wel te zien welke regels zijn toegepast en hoe de regels zijn toegepast, maar niet in welke volgorde. Deze boom kan dan ook bij een andere afleiding horen, bijvoorbeeld:



Toegepaste regel:

(Z,βAB)

(B,βCα)

(A,γαA)

(C,γ)

(A,αα)

Het eindresultaat is hetzelfde, maar de symboolrijen die in de tussenstappen optreden, zien er heel anders uit. We wensen nu deze twee afleidingen als niet essentieel verschillend te beschouwen. Dit betekent, dat we de hierboven afgebeelde ingekorte boom als datgene opvatten, waarin alle essentiële informatie betreffende de afleiding is weergegeven. Onder meer volgt daaruit ook het afgeleide woord βγαααβγα . We willen een dergelijke afleidingsboom nu ook formeel beschrijven. Voordat we dit doen willen we nog op een verschijnsel wijzen. Hierboven hebben we gezien, dat bij verschillende afleidingen dezelfde afleidingsboom kan behoren. Het is echter ook mogelijk dat bij verschillende afleidingsbomen dezelfde afleiding hoort. In een afleidingsstap $X_j \Rightarrow X_{j+1}$ wordt een afleidingsregel toegepast: $X_j = rAs$, $X_{j+1} = rqs$, (A,q) een regel. De vraag is of het mogelijk is om in X_j op een andere plaats een regel toe te passen zo, dat dezelfde X_{j+1} wordt verkregen. Als voorbeeld nemen we een regel (A,Λ) en een overgang $rA^ks \Rightarrow rA^{k-1}s$. Als $k > 1$, is niet vast te stellen op welke A de regel is toegepast. Een ander voorbeeld krijgen we als we beschikken over regels (A,Aαβ) en (B,βαB). Nu gaat rAαBs met de eerste regel over in rAαβαBs, maar met de tweede regel eveneens. Aan deze voorbeelden zien we, dat bij gegeven afleiding de plaats, waar de regel bij een stap wordt toegepast, niet hoeft vast te liggen; bij zulk een afleiding behoren verschillende afleidingsbomen.

Het is mogelijk om aan de regels beperkingen op te leggen, waardoor bovengenoemd verschijnsel onmogelijk wordt gemaakt. Als voor een regel (A,q) geldt $q \neq \Lambda$ en het voorste en het achterste symbool van q zijn $\neq A$, dan ligt bij toepassing van deze regel de plaats waar een regel wordt toegepast vast. Immers, stel $rAs \Rightarrow rqs$ en hetzelfde effect kan worden verkregen door toepassing van een regel op een andere plaats. Als die plaats in r valt, blijft het gedeelte rechts van die plaats onveranderd in strijd met het feit dat het achterste symbool van $q \neq A$ is. Analoog als de plaats in s valt. Voor de beoogde toepassingen is het echter niet gewenst om de genoemde beperking op te leggen. Men noemt regels van de gedaante (A,Ar) <u>linksrecursief</u> en regels van de gedaante (A,rA) <u>rechtsrecursief</u>; dergelijke regels en ook regels met leeg rechterlid komen in toepassingen veel voor.

Men kan echter het verschijnsel ook bestrijden door het begrip afleiding te verfijnen en in de afleiding zelf aan te geven, waar de regel wordt toegepast. Dit leidt tot een gespecificeerde afleiding. Ter inleiding schrijven we een afleiding als volgt:

$Z \Rightarrow r_0 = u_1A_1v_1 \Rightarrow u_1r_1v_1 = u_2A_2v_2 \Rightarrow u_2r_2v_2 \Rightarrow \dots \Rightarrow u_{k-1}r_{k-1}v_{k-1} = u_kA_kv_k \Rightarrow$
$\Rightarrow u_kr_kv_k$ met toepassing van de regels (Z,r_0), $(A_1r_1),\dots,(A_k,r_k)$. Om de eerste stap geen uitzonderingspositie te geven voeren we nog in $A_0 = Z$, $u_0 = v_0 = \Lambda$. Verder geldt $u_kr_kv_k \in I_T^*$. Dit leidt nu tot de volgende formele definitie.

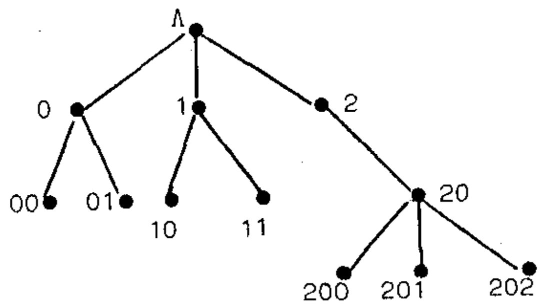
<u>Definitie gespecificeerde afleiding</u>

Stel een contekstvrije grammatica $G = \langle I,I_T,Z,R\rangle$. Een gespecificeerde afleiding bij G bestaat uit $k \in \mathbb{N}$, $u_0,\dots,u_k$, $v_0,\dots,v_k$, alle $\in I^*$,

$(A_0,r_0),\dots,(A_k,r_k)$, alle $\in R$,
$A_0 = Z$, $u_0 = v_0 = \Lambda$, $u_k r_k v_k \in I_T^*$,
$u_j r_j v_j = u_{j+1} A_{j+1} v_{j+1}$ voor $j \in \{0,\dots,k-1\}$.

Bij een gespecificeerde afleiding behoort een afleiding $X_0,\dots,X_{k+1}$, waarin $X_0 = Z$, $X_{j+1} = u_j r_j v_j$ voor $j \in \{0,\dots,k\}$. Het woord $u_k r_k v_k$ heet het <u>afgeleide woord</u> van de gespecificeerde afleiding.

We kunnen bij een afleiding de keuze van het te vervangen hulpsymbool vastleggen door de afspraak steeds het meest linksstaande hulpsymbool te kiezen. De gespecificeerde afleiding wordt dan <u>linkspreferent</u> genoemd; aan de eisen voor een gespecificeerde afleiding wordt dan toegevoegd: $u_0,\dots,u_k$ alle $\in I_T^*$. We verwachten dan een bijectieve betrekking tussen linkspreferente gespecificeerde afleidingen en afleidingsbomen. De laatste gaan we nu definiëren. Een boom is een speciaal geval van een graaf en bestaat uit <u>knopen</u> en <u>takken</u>. In ons geval is er een knoop, waar de afleiding begint, die we de <u>wortel</u> van de boom kunnen noemen. Bij iedere knoop hoort een symbool, maar hetzelfde symbool kan meermalen voorkomen. We kunnen de symbolen dus niet als namen voor de knopen gebruiken; de knopen zijn echter wel <u>gemerkt</u> (Engels: labeled) met symbolen. Uitgaande van de wortel zijn de knopen te verdelen in horizontale lagen, waarin ook een volgorde van links naar rechts is vastgelegd. Dit beschrijven we door de knopen te coderen met rijen natuurlijke getallen, waarbij de met een knoop met code r verbonden volgende knopen als code krijgen r gevolgd door 0,1,... . De wortel geven we de lege rij Λ. Let wel dat deze codering ook de takken van de boom vastlegt. Voor de formele beschrijving noemen we $\mathbb{N}^*$ de verzameling van de eindige rijen natuurlijke getallen



inclusief de lege rij Λ. We schrijven zulke rijen met een komma als scheidingssymbool, bijvoorbeeld 2,6,13,4. Ook voor de concatenatie van zulke rijen gebruiken we een komma: als r en s zulke rijen zijn, dan is r,s de concatenatie van beide. Voor een afleidingsboom is er nog een functie f nodig, die vastlegt welke symbolen aan de knopen worden gehecht; tenslotte moet de vertakking bij iedere knoop corresponderen met de toepassing van regels uit de grammatica en moet de boom een uit eindsymbolen bestaand woord opleveren.

Allereerst definiëren we nog twee afbeeldingen. Als B een eindige deelverzameling van $\mathbb{N}^*$ is definiëren we $D_B : \mathbb{N}^* \to P(\mathbb{N})$ en $v_B : \mathbb{N}^* \to \mathbb{N}$ door voor $t \in \mathbb{N}^*$ te stellen $D_B(t) := \{n \in \mathbb{N} \mid t,n \in B\}$ en $v_B(t) := \# D_B(t)$.

Definitie afleidingsboom

Stel een contekstvrije grammatica $G = \langle I, I_T, Z, R \rangle$. Dan heet $T = \langle B, f \rangle$ met $B \subset \mathbb{N}^*$ en $f : B \to I$ een afleidingsboom als geldt:

1. B eindig, $B \neq \emptyset$,
2. als $t \in \mathbb{N}^*$, $n \in \mathbb{N}$ en $t,n \in B$, dan $t \in B$,
3. als $t \in B$, dan $D_B(t) = \{n \in \mathbb{N} \mid n < v_B(t)\}$,
4. $f(\Lambda) = Z$,
5. als $t \in B$, $f(t) \in I_T$, dan $v_B(t) = 0$,
6. als $t \in B$, $f(t) \in I_N$, dan $(f(t), f(t,0) \ldots f(t, v_B(t) - 1)) \in R$.

We maken enkele opmerkingen over deze definitie. Als $t \in \mathbb{N}^*$, $s \in \mathbb{N}^*$ en $t,s \in B$, dan $t \in B$. Dit volgt met recurrentie langs de opbouw van s uit voorwaarde 2. We noemen t een <u>antecedent</u> van t,s, mits $s \neq \Lambda$. Omdat $B \neq \emptyset$, geldt dus ook $\Lambda \in B$; dit zorgt er voor dat $f(\Lambda)$ in 4. gedefinieerd is. Voorwaarde 3. garandeert dat de in 6. voorkomende toepassingen van f gedefinieerd zijn. Als $t \in B$ en u een antecedent van t, dan $f(u) \in I_N$. Immers, er is een $s \neq \Lambda$ zo, dat $t = u,s$, dus er is een $j \in \mathbb{N}$ zo, dat u,j een antecedent van t of t is. In ieder geval is $u,j \in B$ en $u \in B$; nu levert voorwaarde 5., dat $f(u) \in I_N$. Het is toegestaan, dat als $f(t) \in I_N$ de $v_B(t) = 0$; in dat geval onstaat er in 6. een regel met een leeg rechterlid: $(f(t),\Lambda)$.

Soms geeft men een wat zwakkere definitie, waarbij men niet de eis stelt dat de knopen, die met een gegeven knoop door een dalende tak zijn verbonden, opeenvolgende nummers krijgen: men laat daarin lacunes toe. Dit betekent, dat voorwaarde 3. vervalt, maar dit leidt ertoe dat voorwaarde 6. anders geformuleerd moet worden. Daartoe hebben we de behoefte aan een <u>rangschikkingsfunctie</u>, die we ook voor andere doeleinden kunnen gebruiken. We nemen daarbij als bekend aan, dat men een eindige totaal geordende verzameling zo kan nummeren, dat de ordening van de nummers correspondeert met de ordening in de verzameling. Formeel: als $\langle D, < \rangle$ een eindige totaal geordende verzameling is, dan bestaat er één en slechts één bijectie

$r_D : \{n \in \mathbb{N} \mid n < \#D\} \to D$ zo, dat als $k \in \mathbb{N}, \ell \in \mathbb{N}$, $k < \ell < \#D$, dan $r_D(k) < r_D(\ell)$. In het bijzonder geldt dit als $D \subset \mathbb{N}$, D eindig. De afbeelding r_D hangt behalve van de verzameling D ook van de ordening van D af.

We vervangen 6. nu door

6'. als $t \in B$, $f(t) \in I_N$, dan $(f(t), f(t, r_{D_B(t)}(0)) \ldots f(t, r_{D_B(t)}(v_B(t)-1))) \in R$.

Als we in de definitie van afleidingsboom de zes voorwaarden vervangen door 1.,2.,4.,5.,6'. dan spreken we van een quasi-afleidingsboom. Het is mogelijk om een quasi-afleidingsboom om te zetten in een afleidingsboom; daarbij verandert het straks te definiëren afgeleide woord niet. Dit gaan we niet na.

In $\mathbb{N}^*$ voeren we de bekende lexicografische ordening in. Voor $r \in \mathbb{N}^*$, $s \in \mathbb{N}^*$ geldt $r < s$ dan en slechts dan als:

$$\exists_{t\in\mathbb{N}^*}\ \exists_{u\in\mathbb{N}^*}\ \exists_{v\in\mathbb{N}^*}\ \exists_{i\in\mathbb{N}}\ \exists_{j\in\mathbb{N}}\,[r = t,i,u \text{ en } s = t,j,v \text{ en } i < j]$$

of

$$\exists_{t\in\mathbb{N}^*}\,[t \neq \Lambda \text{ en } s = r,t]\ .$$

Dit is een totale ordening van $\mathbb{N}^*$. Om het afgeleide woord van een afleidingsboom te vinden, nemen we de verzameling $E := \{t \in B \mid f(t) \in I_T\}$ en vormen een woord door de labels van de knopen in E achter elkaar te zetten corresponderend met de lexicografische volgorde van de elementen van E: $f(r_E(0))\ldots f(r_E(\#E - 1))$. Het zal blijken, dat dit woord in L(G) zit.

§7. AFLEIDINGEN EN AFLEIDINGSBOMEN

We leggen nu een verband tussen afleidingsbomen en gespecificeerde afleidingen. We beschrijven daartoe twee constructies. Constructie I maakt bij een gegeven afleidingsboom een linkspreferente gespecificeerde afleiding, waarbij hetzelfde afgeleide woord behoort. Constructie II maakt bij een gegeven gespecificeerde afleiding een afleidingsboom, waarbij hetzelfde afgeleide woord behoort. We tonen echter ook nog aan, dat de twee constructies in zekere zin elkaars inverse zijn. Passen we op een afleidingsboom constructie I toe en op de daaruit verkregen afleiding constructie II, dan krijgen we de gegeven afleidingsboom terug. Gaan we uit van een linkspreferente gespecificeerde afleiding, passen we daarop constructie II toe en op het resultaat daarvan constructie I dan komt er de gegeven afleiding uit.

Het uit te voeren programma kan worden weergegeven in het volgende schema.

afleidingsboom $\xrightarrow{\text{constructie I}}$ linkspref. gespec.afl.(zelfde afgeleide woord)

gespec.afl. $\xrightarrow{\text{constructie II}}$ afleidingsboom (zelfde afgel.woord)

afl. boom $\xrightarrow{\text{I}}$ gespec.afl. $\xrightarrow{\text{II}}$ afl. boom (zelfde afl. boom)

linkspref.gespec.afl. $\xrightarrow{\text{I}}$ afl.boom $\xrightarrow{\text{II}}$ gespec.afl. (zelfde gespec. afleiding)

Als dit voltooid is, is het onverschillig of men met een afleiding dan wel met een afleidingsboom werkt, omdat men desgewenst van de ene op de andere kan overstappen. De keuze geschiedt op pragmatische gronden.

We gaan nu constructie I beschrijven. Er is dan een afleidingsboom gegeven en we moeten komen tot een gespecificeerde afleiding $u_0, \dots$. De letters, die in die woorden voorkomen, worden verkregen door de labels te nemen van knopen uit B. Als tussenstap voeren we daarom rijen knopen $\overline{u_0}, \dots$ in, dat zijn in feite elementen van $(\mathbb{N}^*)^*$. Om verwarring te voorkomen gebruiken we bij de laatste rijvorming de puntkomma als scheidingssymbool.
Voorbeeld: 0,2,1;1,0;3,1,2;1;Λ;2,3. Wat tussen de puntkomma's staat zijn elementen van $\mathbb{N}^*$; deze zullen we in het vervolg in oneigenlijke zin ook letters noemen. Ook de concatenatie in $(\mathbb{N}^*)^*$ wordt met puntkomma aangeduid. Er is nog een moeilijkheid met het lege woord. In $\mathbb{N}^*$ zit het lege woord Λ dat dus als letter in een rij uit $(\mathbb{N}^*)^*$ kan worden gebruikt, zoals trouwens in het gegeven voorbeeld ook gebeurd is. Daarnaast is er ech-

ter ook het lege woord in $(\mathbb{N}^*)^*$; ter onderscheiding zullen we dat met $\underline{\Lambda}$ aanduiden.

Laat nu een afleidingsboom $\langle B,f\rangle$ en dus ook een contekstvrije grammatica gegeven zijn. Definieer $n_0 := v_B(\Lambda)$, dan $n_0 \in \mathbb{N}$ en $\overline{r_0} := 0;\ldots;n_0 - 1$, dan $\overline{r_0} \in B^*$. Verder $\overline{A_0} := \Lambda$, $\overline{u_0} := \underline{\Lambda}$, $\overline{v_0} := \underline{\Lambda}$. We gaan nu een inductief proces toepassen, waarvan we de inductiestap beschrijven. Stel $j \in \mathbb{N}$, $j > 0$ en $\overline{u_{j-1}}$, $\overline{r_{j-1}}$, $\overline{v_{j-1}}$, $\overline{A_{j-1}}$, n_{j-1} verkregen zo, dat $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} \in B^*$, $\overline{A_{j-1}} \in B$, $f(\overline{A_{j-1}}) \in I_N$, $n_{j-1} = v_B(\overline{A_{j-1}})$ en voor alle letters t in $\overline{u_{j-1}}$ geldt $f(t) \in I_T$. Merk op, dat dit bij het begin van het proces voor $j = 1$ geldig is. Als voor alle letters t in $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}}$ geldt, dat $f(t) \in I_T$ dan stellen we $k := j-1$ en breken het proces af. Zo niet, laat dan $\overline{A_j}$ de meest linkse letter t in $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}}$ zijn, waarvoor $f(t) \in I_N$ en maak de concatenatie $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} = \overline{u_j}; \overline{A_j}; \overline{v_j}$. Dit definieert $\overline{u_j}$ en $\overline{v_j}$. Verder definiëren we $n_j := v_B(\overline{A_j})$ en $\overline{r_j} := \overline{A_j},0;\ldots;\overline{A_j},n_j-1$. Dit alles voldoet aan de voorwaarden van de inductiestap met j vervangen door j+1.

Bij een inductiestap kan het proces afbreken of doorgaan. We willen aantonen dat het proces na eindig veel stappen afbreekt. We doen dit door te laten zien dat steeds "nieuwe" knopen van de boom worden gebruikt. In het bijzonder blijken de $\overline{A_j}$ verschillend te zijn. Om dit te bereiken hebben we enkele hulpstellingen nodig, die zelf weer met inductie worden aangetoond. We zullen deze hulpstellingen echter later nogmaals willen gebruiken en maken ze daarom los van de constructie, waar we nu mee bezig zijn. We formuleren eerst een aantal voorwaarden voor de drie hulpstellingen gezamenlijk; deze voorwaarden zijn bij de nu in het geding zijnde constructie vervuld. Deze voorwaarden luiden:

j doorloopt $\mathbb{N}$ of de natuurlijke getallen $\leqslant k$;

$n_j \in \mathbb{N}$, $\overline{A_j} \in \mathbb{N}^*$, $\overline{r_j}$ en $\overline{u_j}$ en $\overline{v_j}$ alle $\in (\mathbb{N}^*)^*$;

$\overline{A_0} = \Lambda$, $\overline{u_0} = \overline{v_0} = \underline{\Lambda}$;

$\overline{r_j} = \overline{A_j},0; \ldots ; \overline{A_j}, n_j-1$;

voor $j \neq 0$: $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} = \overline{u_j}; \overline{A_j}; \overline{v_j}$.

Hulpstelling 7.1. In $\overline{u_j}; \overline{r_j}; \overline{v_j}$ zijn de letters lexicografisch opklimmend geordend; geen der letters in de rij is antecedent van een letter in die rij.

Bewijs: Volledige inductie naar j; $j = 0$ is triviaal. Stel nu $j > 0$ en de bewering juist voor $j-1$. Er geldt nu

$$(*) \qquad \begin{cases} \overline{u_j}; \overline{r_j}; \overline{v_j} = \overline{u_j}; \overline{A_j},0;\ldots; \overline{A_j}, n_j-1; \overline{v_j} , \\ \overline{u_j}; \overline{A_j}; \overline{v_j} = \overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} . \end{cases}$$

De inductiestap is nu eenvoudig uit te voeren. ∎

In de volgende hulpstellingen zijn er i en j, die beide aan de hierboven geformuleerde voorwaarden voor j voldoen.

Hulpstelling 7.2. $\overline{A_i}$ en alle antecedenten van $\overline{A_i}$ komen niet als letter voor in $\overline{u_j}; \overline{r_j}; \overline{v_j}$ voor alle $j \geqslant i$. Als $i \neq 0$ dan komt $\overline{A_i}$ als letter voor in $\overline{u_{i-1}}; \overline{r_{i-1}}; \overline{v_{i-1}}$.

Bewijs: De tweede bewering is triviaal. De eerste bewering bewijzen we met volledige inductie naar j. Voor $j = i$ gaan we uit van $\overline{u_i}; \overline{r_i}; \overline{v_i} = \overline{u_i}; \overline{A_i},0;\ldots; \overline{A_i}, n_i-1; \overline{v_i}$. Als $n_i > 0$ dan volgt de bewering direct uit hulpstelling 7.1. Als echter $n_i = 0$, dan is $\overline{r_i} = \underline{\Lambda}$. Als nu $i > 0$, dan volgt de bewering uit $\overline{u_i}; \overline{A_i}; \overline{v_i} = \overline{u_{i-1}}; \overline{r_{i-1}}; \overline{v_{i-1}}$ en hulpstelling 7.1.

Als echter i = 0, dan $\overline{u_i} = \overline{v_i} = \underline{\Lambda}$ en de bewering is triviaal. We stellen nu $j > i$ en nemen aan dat de bewering voor j-1 geldig is. De inductiestap is nu eenvoudig uit te voeren met de in het bewijs van hulpstelling 7.1 opgeschreven formules (*). ∎

<u>Hulpstelling 7.3</u>. Als $i \neq j$, dan $\overline{A_i} \neq \overline{A_j}$.

<u>Bewijs</u>: Stel $i < j$, dan levert hulpstelling 7.2:

$\overline{A_i}$ komt niet als letter voor in $\overline{u_{j-1}}$; $\overline{r_{j-1}}$; $\overline{v_{j-1}}$,

$\overline{A_j}$ komt wel als letter voor in $\overline{u_{j-1}}$; $\overline{r_{j-1}}$; $\overline{v_{j-1}}$.

Dus $\overline{A_i} \neq \overline{A_j}$. ∎

We keren terug tot het constructieproces. Uit hulpstelling 7.3, uit het feit dat alle $\overline{A_j}$ knopen van B zijn en B eindig is, volgt dat het proces afbreekt en ons oplevert een $k \in \mathbb{N}$ en $\overline{u_0},\ldots,\overline{u_k},\overline{v_0},\ldots,\overline{v_k}$, $\overline{r_0},\ldots,\overline{r_k}$, $\overline{A_0},\ldots,\overline{A_k}$. We krijgen nu de overeenkomstige $u_0,\ldots$ zonder overstreping door letter voor letter de afbeelding f toe te passen. Dat dit een linkspreferente gespecificeerde afleiding oplevert is eenvoudig na te gaan. We moeten nog aantonen dat $u_k r_k v_k$ het afgeleide woord van de afleidingsboom is. In $\overline{u_k}$; $\overline{r_k}$; $\overline{v_k}$ zijn de letters lexicografisch geordend (hulpstelling 7.1) en hun labels zijn eindsymbolen op grond van de constructie. We zijn klaar, als we aantonen, dat als $t \in B$, $f(t) \in I_T$, dan komt t als letter voor in $\overline{u_k}$; $\overline{r_k}$; $\overline{v_k}$. Dit volgt uit hulpstelling 7.4, die wel afhangt van de beschreven constructie.

<u>Hulpstelling 7.4</u>. Stel $t \in B$, $t \notin \{\overline{A_0},\ldots,\overline{A_k}\}$. Als $j \in \mathbb{N}$, $j \leq k$, dan komt t of een antecedent van t als letter voor in $\overline{u_j}$; $\overline{r_j}$; $\overline{v_j}$.

<u>Bewijs</u>: We passen volledige inductie naar j toe; eerst $j = 0$. Omdat $t \neq \overline{A_0} = \Lambda$, is er een $h \in \mathbb{N}$ zo, dat h antecedent van t of t is. Omdat $t \in B$, geldt $h \in B$, $h < v_B(\Lambda)$, dus h komt als letter voor in $\overline{r_0} = \overline{u_0};\overline{r_0};\overline{v_0}$. Stel nu $j > 0$ en de bewering juist voor $j-1$. We gebruiken weer de formules (*) uit het bewijs van hulpstelling 7.1. Als t of een antecedent van t als letter in $\overline{u_j}$ of $\overline{v_j}$ zit zijn we klaar. Als dat echter niet zo is, volgt uit $t \neq \overline{A_j}$ en de inductieveronderstelling dat $\overline{A_j}$ een antecedent is van t. Dan is er een $i \in \mathbb{N}$ zo, dat $\overline{A_j},i$ een antecedent van t of t is. Maar dan geldt $\overline{A_j},i \in B$, dus $i < v_B(\overline{A_j}) = n_j$, dus $\overline{A_j},i$ komt als letter voor in $\overline{r_j}$ en dus in $\overline{u_j};\overline{r_j};\overline{v_j}$. ∎

Uit hulpstelling 7.4 volgt, dat als $t \in B$ en $f(t) \in I_T$, t of een antecedent van t als letter voorkomt in $\overline{u_k};\overline{r_k};\overline{v_k}$, maar voor een antecedent s van t geldt $f(s) \in I_N$, hetgeen niet geldt voor de letters, die in $\overline{u_k};\overline{r_k};\overline{v_k}$ voorkomen. Het is dus t zelf die daarin voorkomt. Hiermee is constructie I tot een goed einde gebracht. Voor later gebruik tonen we nog aan dat $\overline{A_0},\ldots,\overline{A_k}$ alle knopen van B opleveren, waar een hulpsymbool bijhoort.

<u>Hulpstelling 7.5</u>. Stel $t \in B$, dan geldt

$$(1) \qquad f(t) \in I_N \Leftrightarrow \forall_{j \in \mathbb{N}} [j \leq k \Rightarrow t \neq \overline{A_j}].$$

<u>Bewijs</u>. $\Leftarrow$ is triviaal. Stel nu $f(t) \in I_N$ en $\forall_{j\in\mathbb{N}}[j \leqslant k \Rightarrow t \neq \overline{A_j}]$. Nu levert hulpstelling 7.4, dat t of een antecedent van t als letter voorkomt in $\overline{u_k};\overline{r_k};\overline{v_k}$. Dit is echter in strijd met $f(t) \in I_N$ en $f(s) \in I_N$ voor alle antecedenten s van t. ∎

We gaan nu constructie II beschrijven. Daartoe willen we definiëren wat de j^e letter in een woord is, zowel voor de woorden in I^* als die in $(\mathbb{N}^*)^*$ met betrekking tot ; als scheidingssymbool.

Stel S een verzameling symbolen; we definiëren $s_j(x)$ voor $x \in S^*$, $j \in \mathbb{N}$, $j < \ell(x)$ door recurrentie langs de opbouw van x:

Stel $x \in S^*$, $\xi \in S$, dan definiëren we

$$s_j(x\xi) := \begin{cases} s_j(x) \text{ als } j < \ell(x), \\ \xi \quad \text{ als } j = \ell(x). \end{cases}$$

Let wel dat voor Λ niets gedefinieerd hoeft te worden; $s_0(\xi) = \xi$, $s_0(\xi\eta) = \xi$, $s_1(\xi\eta) = \eta$, enz.

Bij constructie II is een gespecificeerde afleiding gegeven, die niet links-preferent hoeft te zijn. Het doel is het definiëren van een afleidingsboom. Als tussenstap gaan we weer overstreepten $\overline{u_0}, \ldots$ invoeren, dat zijn elementen van $(\mathbb{N}^*)^*$ van dezelfde lengte als de overeenkomstige $u_0, \ldots$. Dit gaat als volgt.

$$\overline{A_0} := \Lambda,\ \overline{u_0} := \underline{\Lambda},\ \overline{v_0} := \underline{\Lambda},\ \overline{r_0} := 0; \ldots; \ell(r_0) - 1.$$

We doen een inductiestap. Stel $j \in \mathbb{N}$, $0 < j \leqslant k$ en $\overline{A_{j-1}}$, $\overline{u_{j-1}}$, $\overline{v_{j-1}}$, $\overline{r_{j-1}}$ reeds gedefinieerd en van de goede lengte. Er geldt $u_{j-1}r_{j-1}v_{j-1} = u_jA_jv_j$. Omdat de lengtes bij de index j-1 goed zijn, kunnen we stellen $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} = \overline{u_j}; \overline{A_j}; \overline{v_j}$ en daarmee $\overline{u_j}$ en $\overline{A_j}$ en $\overline{v_j}$ van goede lengte definiëren. Daarna definiëren we $\overline{r_j} := \overline{A_j}, 0; \ldots; \overline{A_j}, \ell(r_j) - 1$. Daarmee zijn de overstreepten gedefinieerd. De definitie garandeert, dat ze voldoen aan de voorwaarden van hulpstellingen 7.1, 7.2 en 7.3 met $n_j = \ell(r_j)$ dus ook aan die hulpstellingen zelf. In het bijzonder zijn de $\overline{A_0}, \ldots, \overline{A_k}$ verschillend.

We gaan nu de boom $\langle B,f\rangle$ definiëren. In B nemen we op Λ en alle letters, die in $\overline{r_0},\ldots,\overline{r_k}$ voorkomen. Dan geldt dat alle letters die in $\overline{u_0},\ldots,\overline{u_k}$, $\overline{v_0},\ldots,\overline{v_k}$ voorkomen in B zitten en dat $\overline{A_0},\ldots,\overline{A_k}$ in B zitten. Dit is eenvoudig met volledige inductie in te zien. De letters die in verschillende $\overline{r_j}$ voorkomen zijn echter verschillend, en verschillend van Λ. Immers, stel dat t voorkomt in $\overline{r_i}$ en u in $\overline{r_j}$, dan zijn er $h \in \mathbb{N}$ en $m \in \mathbb{N}$ zo, dat $t = \overline{A_i},h$ en $u = \overline{A_j},m$. In ieder geval is dan $t \neq \Lambda$ en $t = u$ leidt tot $\overline{A_i} = \overline{A_j}$ en dus tot $i = j$. Op grond hiervan en van het feit dat een letter ten hoogste één maal in $\overline{r_j}$ voorkomt, is de volgende definitie van f gerechtvaardigd.

$$\left.\begin{array}{l} f(\Lambda) := Z \\ f(s_i(\overline{r_j})) = s_i(r_j) \text{ voor } i < \ell(r_j),\ j \leqslant k \end{array}\right\} .$$

Dan geldt echter ook

$$\left.\begin{array}{l} f(s_i(\overline{u_j})) = s_i(u_j) \text{ voor } i < \ell(u_j) \\ f(s_i(\overline{v_j})) = s_i(v_j) \text{ voor } i < \ell(v_j) \\ f(\overline{A_j}) = A_j \end{array}\right\} \quad \text{voor } j \leqslant k .$$

Dit bewijst men met volledige inductie naar j. Daarbij levert $j = 0$ alleen $f(\overline{A_0}) = f(\Lambda) = Z = A_0$. Stel nu $j > 0$ en de bewering waar voor $j-1$. Dan is $f(s_i(\overline{u_j};\overline{A_j};\overline{v_j}) = f(s_i(\overline{u_{j-1}};\overline{r_{j-1}};\ \overline{v_{j-1}})) = s_i(u_{j-1}r_{j-1}v_{j-1}) = s_i(u_jA_jv_j)$. Op grond van de goede lengtes geldt dan ook $f(s_i(\overline{u_j})) = s_i(u_j)$, $f(\overline{A_j}) = A_j$ en $f(s_i(\overline{v_j})) = s_i(v_j)$.

We tonen nu aan dat $\langle B,f\rangle$ een afleidingsboom is. Voorwaarden 1. en 4. zijn triviaal. Stel nu $t \in \mathbb{N}^*$, $n \in \mathbb{N}$ en $t,n \in B$, dan is $t,n \neq \Lambda$, dus is er een $j \in \mathbb{N}$ zo, dat t,n als letter voorkomt in $\overline{r_j}$. Daaruit volgt direct dat $t = \overline{A_j}$ en $n < \ell(r_j)$. Hieruit volgen voorwaarden 2., 3. en 5. Verder volgt eruit dat $v_B(\overline{A_j}) = \ell(r_j)$. We tonen nu aan dat als $t \in B$ en $f(t) \in I_N$ er een $j \in \mathbb{N}$ bestaat zo, dat $t = \overline{A_j}$. Als namelijk $t = \Lambda$, dan $t = \overline{A_0}$. Als echter $t \neq \Lambda$, dan bestaat er een $h \in \mathbb{N}$, $1 \leq h \leq k$, zo, dat t als letter voorkomt in $\overline{u_{h-1}}; \overline{r_{h-1}}; \overline{v_{h-1}}$ en niet als letter voorkomt in $\overline{u_h}; \overline{r_h}; \overline{v_h}$. Dit is zo, omdat, wegens $f(t) \in I_N$, t niet als letter voorkomt in $\overline{u_k};\overline{r_k};\overline{v_k}$ en wel in één der $\overline{r_j}$. Omdat $\overline{u_{h-1}}; \overline{r_{h-1}}; \overline{v_{h-1}} = \overline{u_h}; \overline{A_h}; \overline{v_h}$ volgt hieruit dat $t = \overline{A_h}$.

Stel nu $t \in B$, $f(t) \in I_N$ en kies j zo, dat $t = \overline{A_j}$. Dan geldt

$$(f(t),f(t,0)\ldots f(t,v_B(t)-1)) = (f(\overline{A_j}),f(s_0(\overline{r_j}))\ldots f(s_{\ell(r_j)-1}(\overline{r_j}))) =$$

$$= (A_j,s_0(r_j)\ldots s_{\ell(r_j)-1}(r_j)) = (A_j,r_j) \in R.$$

Daarmee is ook voorwaarde 6. aangetoond. Dus $\langle B,f \rangle$ is een afleidingsboom. We moeten nu nog aantonen dat het afgeleide woord hetzelfde is. Bij de gespecificeerde afleiding is het $u_k r_k v_k$. Beschouw $\overline{u_k}; \overline{r_k}; \overline{v_k}$. Hierin zijn de letters lexicografisch geordend (hulpstelling 7.1); het zijn knopen van de boom en f voert ze over in de overeenkomstige letters van $u_k r_k v_k$, hetgeen elementen van I_T zijn. Het enige dat we nog moeten aantonen is, dat iedere $t \in B$ waarvoor $f(t) \in I_T$, als letter in $\overline{u_k}; \overline{r_k}; \overline{v_k}$ voorkomt. Voor zulk een t geldt $t \neq \Lambda$, dus er is een $h \in \mathbb{N}$ zo, dat t als letter in $\overline{r_h}$ voorkomt. Dan komt t ook als letter voor in $\overline{u_j}; \overline{r_j}; \overline{v_j}$ voor alle $j \in \mathbb{N}$ met $h \leq j \leq k$. Dit bewijzen we met volledige inductie naar j. Voor $j = h$ klopt het. Stel $h < j \leq k$ en de bewering waar voor $j-1$, Nu is $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} = \overline{u_j}; \overline{A_j}; \overline{v_j}$. Wegens $f(t) \in I_T$ is $t \neq \overline{A_j}$. Op grond van de inductieveronderstelling komt t als letter voor in $\overline{u_j}; \overline{v_j}$ en dus in $\overline{u_j}; \overline{r_j}; \overline{v_j}$. Het geval dat $j = k$ is levert de gewenste bewering. Hiermee is constructie II voltooid.

Nu moeten we bij boom $\xrightarrow{\mathrm{I}}$ afleiding $\xrightarrow{\mathrm{II}}$ boom laten zien dat de bomen hetzelfde zijn. We krijgen notatieproblemen: bij beide constructies ontstaan overstreepten en in II ontstaat een nieuwe boom. We lossen dit op door bij II accenten te gebruiken: *we noemen de overstreepten* $\overline{u_0'},\ldots$ en de nieuwe boom $\langle B',f'\rangle$. We bewijzen eerst dat de overstreepten bij beide constructies hetzelfde zijn: $\overline{u_0'} = \overline{u_0}$, enz. Dat $k' = k$ is triviaal. $\overline{r_0} = 0;\ldots;n_0-1$ en $\ell(r_0) = n_0$ en dan $\overline{r_0'} = 0;\ldots;\ell(r_0)-1$, dus $\overline{r_0'} = \overline{r_0}$. Verder $\overline{u_0'} = \underline{\Lambda} = \overline{u_0}$, $\overline{A_0'} = \Lambda = \overline{A_0}$, $\overline{v_0'} = \underline{\Lambda} = \overline{v_0}$. Nu de inductiestap. Stel $0 < j \leqslant k$ en $\overline{u_{j-1}'} = \overline{u_{j-1}}$, $\overline{v_{j-1}'} = \overline{v_{j-1}}$, $\overline{r_{j-1}'} = \overline{r_{j-1}}$, $\overline{A_{j-1}'} = \overline{A_{j-1}}$, dan is $\overline{u_j};\overline{A_j};\overline{v_j} = \overline{u_{j-1}};\overline{r_{j-1}};\overline{v_{j-1}} = \overline{u_{j-1}'};\overline{r_{j-1}'};\overline{v_{j-1}'} = \overline{u_j'};\overline{A_j'};\overline{v_j'}$, maar $\ell(\overline{u_j}) = \ell(u_j) = \ell(\overline{u_j'})$, dus $\overline{u_j'} = \overline{u_j}$, $\overline{A_j'} = \overline{A_j}$, $\overline{v_j'} = \overline{v_j}$ en $\overline{r_j} = \overline{A_j},0;\ldots;\overline{A_j},n_j-1 = \overline{A_j'},0;\ldots;\overline{A_j'},\ell(r_j)-1 = \overline{r_j'}$.

Omdat de overstreepten hetzelfde zijn kunnen we het eerste deel van contructie I uitvoeren en op het resultaat daarvan het tweede deel van constructie II. Triviaal is dan dat $B' \subset B$ en dat voor $t \in B'$ geldt dat $f'(t) = f(t)$. We behoeven dus alleen nog $B \subset B'$ te bewijzen. Daartoe is voldoende dat iedere $t \in B$ als letter in één der overstreepten voorkomt. Als $f(t) \in I_T$ is dat bij constructie I aangetoond en als $f(t) \in I_N$ volgt het uit hulpstelling 7.5.

Nu moeten we nog laten zien dat bij linkspref.afl. $\xrightarrow{\mathrm{II}}$ boom $\xrightarrow{\mathrm{I}}$ afl. de gespecificeerde afleidingen hetzelfde zijn. Notatieverwarring voorkomen we nu door bij constructie I zowel de overstreepten als de gevonden afleiding van accenten te voorzien: $\overline{u_0'},\ldots$ en $u_0',\ldots$. Ook hier tonen we eerst aan dat de overstreepten hetzelfde zijn. Bij constructie II is aangetoond dat $v_B(\Lambda) = v_B(\overline{A_0}) = \ell(r_0)$, dus $\overline{r_0'} = 0;\ldots;\ell(r_0)-1 = \overline{r_0}$. Verder $\overline{u_0'} = \underline{\Lambda} = \overline{u_0}$;

$\overline{v_0'} = \underline{\Lambda} = \overline{v_0}, \overline{A_0'} = \Lambda = \overline{A_0}$. Als $k = 0$ dan $r \in I_T$. Dan geldt voor alle letters t in $\overline{r_0}$ dat $f(t) \in I_T$, dus $k' = 0$. We bewijzen met volledige inductie naar j dat als $0 \leqslant j \leqslant k$ dan $j \leqslant k'$ en $\overline{u_j'} = \overline{u_j}$, $\overline{v_j'} = \overline{v_j}$, $\overline{A_j'} = \overline{A_j}$, $\overline{r_j'} = \overline{r_j}$ en als $j = k$, dan $k' = k$. Voor $j = 0$ klopt dit. Stel nu $0 < j \leqslant k$ en de bewering waar voor $j-1$. Nu is $u_{j-1}r_{j-1}v_{j-1} = u_jA_jv_j$ en $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} = \overline{u_j}; \overline{A_j}; \overline{v_j}$ met overeenkomstige lengtes. De gespecificeerde afleiding is linkspreferent verondersteld en dus geldt voor alle letters t in $\overline{u_j}$, dat $f(t) \in I_T$ en verder $f(\overline{A_j}) = A_j \in I_N$. Verder is $\overline{u_{j-1}}; \overline{r_{j-1}}; \overline{v_{j-1}} = \overline{u_{j-1}'};\overline{r_{j-1}'};\overline{v_{j-1}'}$. Daaruit volgt nu dat $\overline{u_j'} = \overline{u_j}$, $\overline{A_j'} = \overline{A_j}$, $\overline{v_j'} = \overline{v_j}$. Verder is bij constructie II aangetoond dat $v_B(\overline{A_j}) = \ell(r_j)$, dus $v_B(\overline{A_j'}) = \ell(r_j)$ en dus $\overline{r_j'} =$
$= \overline{A_j'},0;\ldots;\overline{A_j'},\ell(r_j)-1 = \overline{A_j},0;\ldots;\overline{A_j},\ell(r_j)-1 = \overline{r_j}$. Tevens is $j \leqslant k'$. Als $j = k$, dan $u_jr_jv_j \in I_T^*$, dat wil zeggen dat voor alle letters t in $\overline{u_j'};\overline{r_j'};\overline{v_j'} = \overline{u_j};\overline{r_j};\overline{v_j}$ geldt $f(t) \in I_T$, waaruit volgt dat $k' = j = k$.

Nu we aangetoond hebben dat de overstreepten hetzelfde zijn, kunnen we het eerste deel van constructie II uitvoeren en op het resultaat het tweede deel van constructie I. Dit levert dezelfde afleiding op. Allereerst is $k' = k$; verder $\ell(u_j') = \ell(\overline{u_j'}) = \ell(\overline{u_j}) = \ell(u_j)$ en $s_i(u_j') = f(s_i(\overline{u_j'})) =$
$= f(s_i(\overline{u_j})) = s_i(u_j)$ voor $i < \ell(u_j)$ dus $u_j' = u_j$. Bij v_j, r_j en A_j gaat het analoog.

Hiermee is het programma dat wederzijdse omzetting van afleidingsbomen en gespecificeerde afleidingen beoogde, voltooid. We gaan dit nu gebruiken bij een uitsnijding uit een afleidingsboom. Stel in een afleidingsboom een knoop u en beschouw alle knopen in de boom, die u als antecedent hebben of u zijn. Als $f(u) \in I_T$ is dat alleen u en dus niet interessant. We beperken ons daarom tot het geval dat $f(u) = A \in I_N$. De verzameling bestaat dan uit knopen , die als rij getallen u als beginstuk hebben. Knippen we er die u aan de voorkant af, dan krijgen we iets dat op een afleidingsboom lijkt, maar waarvan de wortel A als label heeft.

Formeel definiëren we bij een grammatica G en een hulpsymbool A een <u>A-boom</u> als een afleidingsboom behorende bij de grammatica, die uit G ontstaat door het startsymbool door A te vervangen. Gaan we nu uit van een afleidingsboom $\langle B,f\rangle$ bij G, $u \in B$ en $f(u) = A \in I_N$ dan definiëren we $B' := \{t \in \mathbb{N}^* \mid u,t \in B\}$ en $f'(t) := f(u,t)$ voor $t \in B'$. Het is eenvoudig in te zien, dat $\langle B',f'\rangle$ een A-boom bij G is. Verder geldt voor het afgeleide woord x van deze boom, dat $A \overset{*}{\underset{G}{\Rightarrow}} x$. We zullen laten zien dat het bij $\langle B,f\rangle$ behorende afgeleide woord de gedaante yxz heeft. Daarbij gebruiken we de volgende eigenschap van lexicografische ordening. Stel $u \in \mathbb{N}^*$, $t \in \mathbb{N}^*$, $s \in \mathbb{N}^*$, dan geldt:

als $u < s$ en u geen **antecedent** van s, dan $u,t < s$.

De verzameling knopen van B, waarbij een eindsymbool hoort, verdelen we op de volgende wijze in drie klassen:

1. de knopen, die lexicografisch aan u voorafgaan;
2. de knopen, die u als antecedent hebben;
3. de knopen, die lexicografisch op u volgen, maar u niet als antecedent hebben.

Lexicografisch gaan de knopen in 1. vooraf aan die in 2. en die in 2. aan die in 3. Als er respectievelijk de woorden y, x en z bij horen, dan is het afgeleide woord yxz en x het afgeleide woord van de A-boom $\langle B',f'\rangle$.

Veronderstel nu dat we nog een A-boom $\langle B'',f''\rangle$ bij G hebben met afgeleid woord x' en dezelfde A als tevoren. We kunnen nu boomchirurgie bedrijven en de A-boom $\langle B',f'\rangle$ uitsnijden en vervolgens vervangen door de A-boom $\langle B'',f''\rangle$. Dit levert weer een afleidingsboom op met afgeleid woord yx'z. Formeel betekent dit de volgende definitie:

$$B''' := (B\backslash\{u,t \mid t \in \mathbb{N}^*\}) \cup \{u,t \mid t \in B''\} .$$

Let wel dat de twee verzamelingen in het rechterlid disjunct zijn. Dit rechtvaardigt de volgende definitie van $f''' : B''' \to I$:

$$f'''(s) := f(s) \quad \text{als } s \in B\backslash\{u,t \mid t \in \mathbb{N}^*\} ;$$

$$f'''(u,t) := f''(t) \quad \text{als } t \in B'' .$$

Het is gemakkelijk te bewijzen dat $\langle B''', f'''\rangle$ een afleidingsboom is met afgeleid woord yx'z .

§8. OPERATIES OP TALEN. POMPSTELLING

Stelling 8.1. Stel I_T verzameling symbolen, $L \subset I_T^*$, $M \subset I_T^*$. Dan geldt:

als L en M contekstvrij, dan $L \cup M$ en LM contekstvrij ,

als L contekstvrij, dan L^* contekstvrij.

Bewijs: Stel de contekstvrije grammatica's $G_0 = \langle I_0 I_T, Z_0 R_0\rangle$ zo, dat $L = L(G_0)$ en $G_1 = \langle I_1, I_T, Z_1, R_1\rangle$ zo, dat $M = L(G_1)$.
Voor $I_{0N} = I_0\backslash I_T$ en $I_{1N} = I_1\backslash I_T$ mogen we zonder beperking van de algemeenheid veronderstellen dat $I_{0N} \cap I_{1N} = \emptyset$. Als dat namelijk niet zo is vervangen we de hulpsymbolen bij G_1 door andere met overeenkomstige aanpassing van Z_1 en R_1 en zo, dat de gegenereerde taal niet verandert. We kiezen nu een nieuw hulpsymbool, dat we Z noemen. We definiëren nu de grammatica $G_3 = \langle I_{3N} \cup I_T, I_T, Z_3, R_3\rangle$ door $I_{3N} := I_{0N} \cup I_{1N} \cup \{Z\}$, $Z_3 := Z$, $R_3 := R_0 \cup R_1 \cup \{(Z,Z_0),(Z,Z_1)\}$. Dan is G_3 een contekstvrije grammatica;

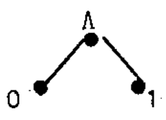
we tonen aan dat $L(G_3) = L \cup M$.

Stel $x \in L \cup M$. Als $x \in L$ dan $Z_0 \overset{*}{\underset{G_0}{\Rightarrow}} x$, dus $Z \underset{G_3}{\Rightarrow} Z_0 \overset{*}{\underset{G_3}{\Rightarrow}} x$. Als $x \notin L$, dan $x \in M$, dus $Z_1 \overset{*}{\underset{G_1}{\Rightarrow}} x$, dus $Z \underset{G_3}{\Rightarrow} Z_1 \overset{*}{\underset{G_3}{\Rightarrow}} x$. In beide gevallen geldt $x \in L(G_3)$. Hieruit volgt $L \cup M \subset L(G_3)$.

Stel omgekeerd $x \in L(G_3)$. De eerste stap in een afleiding van x is $Z \Rightarrow Z_0$ of $Z \Rightarrow Z_1$. In het eerste geval geldt $Z_0 \overset{*}{\underset{G_3}{\Rightarrow}} x$. Voor een afleiding $Z_0 = X_0,\dots,X_k = x$ van x uit Z_0 geldt nu dat voor alle $j \in \{0,\dots,k-1\}$ in X_j uitsluitend hulpsymbolen uit I_{0N} voorkomen en bij de afleiding $X_i \Rightarrow X_{i+1}$ een regel uit R_0 wordt toegepast. Dit bewijst men eenvoudig met inductie naar j; hierbij wordt $I_{0N} \cap I_{1N} = \emptyset$ gebruikt. Hieruit volgt $Z_0 \overset{*}{\underset{G_0}{\Rightarrow}} x$ en dus $x \in L(G_0) \subset L \cup M$. In geval de afleiding met $Z \Rightarrow Z_1$ begint vinden we op analoge wijze $x \in L(G_1) \subset L \cup M$. Daarmee is ook $L(G_3) \subset L \cup M$ aangetoond. Dus $L(G_3) = L \cup M$ en $L \cup M$ is contekstvrij. We definiëren nu de grammatica $G_4 = \langle I_{4N} \cup I_T, I_T, Z_4, R_4 \rangle$ door $I_{4N} := I_{0N} \cup I_{1N} \cup \{Z\}$, $Z_4 := Z$, $R_4 := R_0 \cup R_1 \cup \{(Z, Z_0Z_1)\}$. Dan is G_4 een contekstvrije grammatica; we tonen aan dat $L(G_4) = LM$.

Stel $x \in LM$. Dan zijn er $y \in L$, $z \in M$ zo, dat $x = yz$. Dan $Z_0 \overset{*}{\underset{G_0}{\Rightarrow}} y$ en $Z_1 \overset{*}{\underset{G_1}{\Rightarrow}} z$ en dus $Z \overset{*}{\underset{G}{\Rightarrow}} Z_0Z_1 \overset{*}{\underset{G}{\Rightarrow}} y\ Z_1 \overset{*}{\underset{G}{\Rightarrow}} yz = x$, dus $x \in L(G_4)$. Hieruit volgt $LM \subset L(G_4)$.

Stel omgekeerd $x \in L(G_4)$. Omdat (Z, Z_0Z_1) de enige regel bij G_4 is met Z_4 in het linkerlid ziet de bovenkant van een afleidingsboom voor x bij G_4 er als volgt uit:



met $f(\Lambda) = Z$, $f(0) = Z_0$, $f(1) = Z_1$. De uitsnijding van 0 levert een Z_0-boom met afgeleid woord y en de uitsnijding bij 1 levert een Z_1-boom met afgeleid woord z en $x = yz$. Met inductie naar de lengte van de getalrij bewijst men nu gemakkelijk dat in een Z_0-boom aan iedere knoop een element van I_0 als label is toegevoegd en als bij die knoop een hulpsymbool hoort, daar een regel uit R_0 is toegepast. Daaruit volgt dat een Z_0-boom bij G_4 een afleidingsboom bij G_0 is. Daaruit volgt $y \in L(G_0) = L$. Op analoge wijze toont men aan dat $z \in M$ en dus $x = yz \in LM$. Daarmee is $L(G_4) \subset LM$ aangetoond. Dus $L(G_4) = LM$ en LM is contekstvrij. We definiëren nu de grammatica $G_5 = \langle I_{5N} \cup I_T, I_T, Z_5, R_5 \rangle$ door $I_{5N} := I_{0N} \cup \{Z\}$, $Z_5 := Z$, $R_5 := R_0 \cup \{(Z, ZZ_0), (Z, \Lambda)\}$. Dan is G_5 een contekstvrije grammatica; we tonen aan dat $L(G_5) = L^*$. Stel $x \in L^*$. Dan is er een $n \in \mathbb{N}$, $y_0, \ldots, y_{n-1}$, alle $\in L$ zo, dat $x = y_0 \ldots y_{n-1}$, dus $Z_0 \overset{*}{\underset{G_0}{\Rightarrow}} y_0, \ldots, Z_0 \overset{*}{\underset{G_0}{\Rightarrow}} y_{n-1}$. Als $n = 0$, dan $x = \Lambda$, maar $Z \underset{G_5}{\Rightarrow} \Lambda$, dus $x \in L(G_5)$. Als $n \geqslant 1$, dan

$$Z \underset{G_5}{\Rightarrow} ZZ_0 \underset{G_5}{\Rightarrow} \cdots \underset{G_5}{\Rightarrow} ZZ_0^n \underset{G_5}{\Rightarrow} Z_0^n \overset{*}{\underset{G_5}{\Rightarrow}} y_0 Z_0^{n-1} \overset{*}{\underset{G_5}{\Rightarrow}} \cdots \overset{*}{\underset{G_5}{\Rightarrow}} y_0 \ldots y_{n-1} = x, \text{ dus } x \in L(G_5).$$

Hieruit volgt $L^* \subset L(G_5)$. Stel nu omgekeerd $x \in L(G_5)$. Dan is er een afleidingsboom bij G_5 met afgeleid woord x. We bewijzen $x \in L^*$ met volledige inductie naar het aantal knopen in de afleidingsboom. Voor de regel die bij de wortel van de boom wordt toegepast zijn er twee mogelijkheden. Als dit (Z, Λ) is dan is $x = \Lambda \in L^*$. Als het echter (Z, ZZ_0) is, dan ziet de bovenkant van de boom er als volgt uit:

met $f(\Lambda) = f(0) = Z$ en $f(1) = Z_0$. De uitsnijding bij 0 is een Z-boom met afgeleid woord y, maar een Z-boom is een afleidingsboom. Deze heeft minder knopen dan de gegeven boom; op grond van de inductieveronderstelling is $y \in L^*$. De uitsnijding bij 1 is een Z_0-boom met afgeleid woord z. Op analoge wijze als in het vorige geval bewijst men, dat deze Z_0-boom een afledingsboom bij G_0 is en dus $z \in L(G_0) = L$. Dus $x = yz \in L^* L \subset L^*$. Daarmee is $L(G_5) \subset L^*$ aangetoond. Dus $L(G_5) = L^*$ en L^* is contekstvrij. ∎

Vergelijken we Stelling 8.1 met Stelling 4.1 dan zien we dat nu $L \cap M$ en $I_T^*\backslash L$ ontbreken. Deze behoeven bij contekstvrije talen L en M ook niet contekstvrij te zijn. Neem bijvoorbeeld

$$L:= \{\alpha^n\beta^n\gamma^m | n \in \mathbb{N}, m \in \mathbb{N}\},$$

$$M:= \{\alpha^m\beta^n\gamma^n | n \in \mathbb{N}, m \in \mathbb{N}\},$$

dan zijn deze contekstvrij, want $L = \{\alpha^n\beta^n | n \in \mathbb{N}\}\{\gamma\}^*$ en $M = \{\alpha\}^*\{\beta^n\gamma^n | n \in \mathbb{N}\}$. Verder $L \cap M = \{\alpha^n\beta^n\gamma^n | n \in \mathbb{N}\}$.
Binnenkort zullen we aantonen dat deze taal niet contekstvrij is. Nemen we dit even als waar aan, dan volgt daaruit ook dat niet van iedere taal het complement contekstvrij is. Zou dit namelijk wel zo zijn, dan levert de verzamelingstheoretische gelijkheid

$$L \cap M = I_T^*\backslash((I_T^*\backslash L) \cup (I_T^*\backslash M)),$$

toegepast op bovengenoemde talen L en M op, dat het linkerlid niet en het rechterlid wel contekstvrij is, hetgeen tot een contradictie leidt. Dit levert nog geen expliciet voorbeeld van een contekstvrije taal met een niet-contekstvrij complement. In het rechterlid wordt namelijk drie keer een complement genomen en het is niet duidelijk bij welke complementering het mis gaat.
De volgende stelling bewijzen we niet.

Stelling 8.2. Stel I_T verzameling symbolen, $L \subset I_T^*$, $M \subset I_T^*$. Als L contekstvrij en M regulier is, dan is $L \cap M$ contekstvrij.

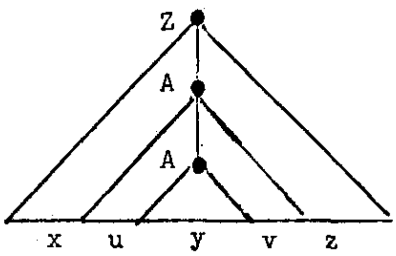
Uit Stelling 8.2 volgt dat als men aan een contekstvrije taal eindig veel woorden toevoegt of er eindig veel elementen uit weglaat, er weer een contekstvrije taal ontstaat. Toevoegen aan een contekstvrije taal L betekent vorming van $L \cup E$ met E eindig, maar een eindige taal is regulier en dus contekstvrij, dus $L \cup E$ is contekstvrij. Weglaten betekent de vorming van $L \cap (I_T^* \setminus E)$ met E eindig, maar omdat E regulier is, is $I_T^* \setminus E$ ook regulier, en dus $L \cap (I_T^* \setminus E)$ contekstvrij.

We leiden nu een eigenschap van contekstvrije talen af, die ons in staat zal stellen van bepaalde talen aan te tonen, dat ze niet contekstvrij zijn. Stel $G = \langle I, I_T, Z, R \rangle$ een contekstvrije grammatica en stel $n := \#I_N$, $m :=$ maximum van de lengtes van de rechterleden van de regels in R. Voor een afleidingsboom bij G geldt dan, dat alle getallen, die in een knoop van de boom voorkomen $< m$ zijn. Verder noemen we bij een afleidingsboom het maximum van de lengtes van de knopen de hoogte van de afleidingsboom.

Hulpstelling 8.3. Als een afleidingsboom bij G met hoogte h een afgeleid woord w heeft, dan $\ell(w) \leqslant m^h$.

Bewijs: We behandelen eerst $m = 0$; dan zijn alle rechterleden van regels leeg, $w = \Lambda$, $\ell(w) = 0 \leqslant 0^h$ (ook geldt $h = 0$). Stel nu verder $m \geqslant 1$. Stel W = de verzameling van alle rijen natuurlijke getallen $< m$ met rijlengte h. Dan $\#W = m^h$. Definieer nu een afbeelding φ: $W \to B$ als volgt: als $r \in W$, dan is $\varphi(r)$ het langste beginstuk van r of r, dat in B zit. Deze definitie is gerechtvaardigd omdat $\Lambda \in B$. We tonen nu aan, dat iedere $t \in B$ met $f(t) \in I_T$ in het beeld van φ voorkomt. De lengte van zo'n t is $\leqslant h$; definieer $r = t,0,\ldots,0$ zo, dat $\ell(r) = h$, dan geldt $r \in W$ (hier wordt $m \geqslant 1$ gebruikt) en $\varphi(r) = t$, want $t \in B$ en ieder langer beginstuk van r of r, dat in B ligt zou t als antecedent hebben, hetgeen niet kan omdat $f(t) \in I_T$. Dus $E \subset \varphi(W)$ en $\ell(w) = \#E \leqslant \#\varphi(W) \leqslant \#W = m^h$. ∎

Stel nu $w \in L(G)$, $\ell(w) > m^n$. Dan is er een afleidingsboom bij G met afgeleid woord w en dus ook zo'n boom met een minimaal aantal knopen; noem deze boom <B,f>. Laat h de hoogte van die boom zijn, dan volgt uit hulpstelling 8.3, dat $n < h$. Kies nu $t \in B$ zo, dat $\ell(t) = h$. De antecedenten van t geven h knopen van B waarbij een hulpsymbool behoort. Wegens $h > n$ zijn er bij die knopen twee verschillende, waarbij hetzelfde hulpsymbool A behoort. Van deze twee is er één antecedent van de andere.



Hierbij horen twee uitsnijdingen, die een A-boom opleveren. Laat de onderste afgeleid woord y hebben, dan heeft het afgeleide woord van de bovenste de gedaante uyv en w = xuyvz. We kunnen nu ook proberen de knopen waar A bijhoort zo laag mogelijk in de boom te kiezen. We nemen dan in de verzameling antecedenten van t de n+1 onmiddellijke voorgangers van t en daaruit twee knopen, waarbij eenzelfde hulpsymbool A behoort. Dan is de hoogte van de uitsnijding bij zo'n knoop $\leqslant n+1$. Uit hulpstelling 8.3 volgt dan dat $\ell(uyv) \leqslant m^{n+1}$.

We gaan nu boomchirurgie bedrijven en snijden de grote A-boom eruit en vervangen deze door de kleine A-boom. Dit levert een afleidingsboom met afgeleid woord xyz, dus $xyz \in L(G)$. Nu geldt $uv \neq \Lambda$. Zou namelijk $uv = \Lambda$ zijn, dan $u = v = \Lambda$, dus $w = xuyvz = xyz$. De zojuist geconstrueerde boom zou dan w als afgeleid woord hebben en minder knopen hebben dan de gegeven boom in strijd met de minimale keuze. We kunnen nu echter ook uit de gegeven boom de kleine A-boom uitsnijden en deze door de grote A-boom vervangen. Dit levert een afleidingsboom met afgeleid woord $xuuyvvz = xu^2yv^2z \in L(G)$. Dit proces kan herhaald worden. Er geldt: als $k \in \mathbb{N}$, dan is er een afleidingsboom bij G met afgeleid woord xu^kyv^kz en deze bevat een uitsnijding, die een A-boom is, die de in genoemde concatenatie voorkomende y als afgeleid woord heeft. Dit ziet men in met volledige inductie naar k. Het geval dat

$k = 0$ is al behandeld. Stel de bewering waar voor k. Snijd de A-boom, die y oplevert, uit en vervang deze door de A-boom die uyv levert. Dan ontstaat een afleidingsboom met afgeleid woord $xu^{k+1}yv^{k+1}z$. Hiermee is de volgende stelling bewezen.

Stelling 8.4. Stel $G = \langle I, I_T, Z, R \rangle$ een contekstvrije grammatica, $n = \# I_N$, m = maximum van de lengtes van de rechterleden van de elementen van R, $w \in L(G)$, $\ell(w) > m^n$. Dan bestaan er x,y,z,u,v, alle $\in I_T^*$ zo, dat

$$w = xuyvz,\ uv \neq \Lambda,\ \ell(uyv) \leqslant m^{n+1},$$

$$\{xu^k yv^k z \mid k \in \mathbb{N}\} \subset L(G).$$

Omdat $\ell(xu^k yv^k z)$ een stijgende functie van k is, is $\{xu^k yv^k z \mid k \in \mathbb{N}\}$ een oneindige verzameling. Hieruit volgt, dat als L(G) eindig is en $w \in L(G)$, dan $\ell(w) \leqslant m^n$.

We kunnen uit Stelling 8.4, die betrekking heeft op een contekstvrije grammatica, direct een stelling maken, die betrekking heeft op een contekstvrije taal en waarin niet expliciet sprake is van een grammatica. Deze stelling is in de literatuur als pompstelling (Engels: pumping lemma) bekend.

Stelling 8.5 (pompstelling). Als L een contekstvrije taal is over een verzameling symbolen I_T, dan bestaat er een $p \in \mathbb{N}$ zo, dat voor iedere $w \in L$ met $\ell(w) > p$ geldt, dat er x,y,z,u,v, alle $\in I_T^*$ bestaan zo, dat

$$w = xuyvz,\ uv \neq \Lambda,\ \ell(uyv) \leqslant p,$$

$$\{xu^k yv^k z \mid k \in \mathbb{N}\} \subset L.$$

Bewijs: Kies een contekstvrije grammatica G met $L = L(G)$ en definieer n en m als in Stelling 8.4. Kies nu $p := m^{n+1}$ en pas Stelling 8.4 toe. ∎

We gebruiken de pompstelling om aan te tonen, dat de taal $\{\alpha^n\beta^n\gamma^n | n \in \mathbb{N}\}$ niet contekstvrij is. Noem deze taal L en veronderstel dat L contekstvrij is. Kies bij L een $p \in \mathbb{N}$ als in de pompstelling en $n > \frac{1}{3}p$, dan $\alpha^n\beta^n\gamma^n \in L$ en $\ell(\alpha^n\beta^n\gamma^n) > p$. De pompstelling levert nu x, y, z, u, v zo, dat $uv \neq \Lambda$ en $\{xu^kyv^kz | k \in \mathbb{N}\} \subset L$. Als nu u twee van de letters α,β,γ bevat, dan bevat $u^2 = uu$ die twee letters in de verkeerde volgorde in strijd met $xu^2yv^2z \in L$. Dus u bevat ten hoogste één van de letters α,β,γ. Voor v geldt op analoge wijze hetzelfde. Er is dus één der letters α,β,γ (noem deze δ), die noch in u, noch in v voorkomt. Laat i het aantal malen zijn, dat δ in xyz voorkomt, dan is dat ook het aantal malen dat δ in xuyvz en in xu^2yv^2z voorkomt. Daaruit en uit $xuyvz \in L$, $xu^2yv^2z \in L$ volgt dan $\ell(xuyvz) = 3i = \ell(xu^2yv^2z)$, maar dit is in strijd met $\ell(xuyvz) < \ell(xu^2yv^2z)$. Dus L is niet contekstvrij. Op te merken is, dat er een contekstgevoelige grammatica bestaat, die L genereert. Daaruit volgt dat $\mathcal{L}_2 \neq \mathcal{L}_1$.

Een ander voorbeeld krijgen we als we de taal $\{w\gamma w | w \in \{\alpha,\beta\}^*\}$ beschouwen. Met behulp van de pompstelling kan bewezen worden dat deze taal niet contekstvrij is, hetgeen we hier niet uitvoeren. Men kan ook aantonen dat het complement van deze taal wel contekstvrij is.

§9. BESLISBAARHEID. AMBIGUÏTEIT

In dit college zullen wij de beslisbaarheid van een taal slechts op intuïtieve wijze behandelen. In §1 is al opgemerkt, dat het begrip beslisbaar of recursief in het college Toegepaste logica I formeel wordt behandeld. Wij beperken ons tot de vaststelling, dat we een taal $L(L \subset I_T^*)$ beslisbaar noemen als er een effectieve procedure bestaat, die voor iedere $x \in I_T^*$ uitmaakt of $x \in L$ dan wel $x \notin L$. Voor toepassingen is het van belang, dat een taal beslisbaar is. Van een programmeertaal verlangen we, dat het verifieerbaar is of een gegeven tekst syntactisch correct is, dat wil zeggen in overeenstemming met de grammaticale voorschriften. In de praktijk eisen we echter vaak meer, namelijk dat van een correcte tekst ook de grammaticale structuur kan worden vastgesteld. Dit laatste correspondeert met een afleidingsboom bij de grammatica.

We noemen een grammatica G beslisbaar als er een effectieve procedure bestaat, die voor iedere $x \in I_T^*$ oplevert: hetzij $x \notin L(G)$, hetzij een afleiding van x bij G. Ingeval G contekstvrij is kan de afleiding door een afleidingsboom worden vervangen, omdat afleidingen en afleidingsbomen wederzijds op effectieve wijze in elkaar kunnen worden omgezet.

Het doel van de nu volgende beschouwingen is om aan te tonen dat een contekstvrije grammatica beslisbaar is. We gaan daarbij uit van de vaststelling dat een niet-dalende grammatica beslisbaar is; dit betekent dat voor alle regels (u,v) van de grammatica geldt $\ell(u) \leq \ell(v)$. Voor een afleiding $Z = X_0, X_1, \ldots, X_k = x$ geldt dan $\ell(X_j) \leq \ell(X_{j+1})$ voor $j \in \{0,\ldots,k-1\}$. Bij een gegeven grammatica is effectief vast te stellen of een rij elementen van I^* wel of niet afleiding van een woord x is. Bovendien kan een afleiding van x vervangen worden door een afleiding zonder herhalingen, omdat men bij een herhaling eenvoudig een deel van de afleiding weg kan laten. Men kan nu effectief een lijst maken van alle rijen woorden zonder herhalingen en met niet-dalende lengtes, die beginnen met Z en eindigen met het gegeven woord x. Van elk van die rijen kan effectief worden vastgesteld of het wel of niet een afleiding is. Op deze wijze wordt effectief vastgesteld of x een afleiding bezit en wordt als dat zo is ook effectief een afleiding gevonden. Daarmee is de beslisbaarheid van een niet-dalende grammatica aangetoond.

Een contekstvrije grammatica hoeft niet niet-dalend te zijn; de storende regels zijn die van de gedaante (A,Λ). Als zulke regels niet voorkomen, is het lege woord niet afleidbaar. Voor de afleidbaarheid van Λ is slechts de regel (Z,Λ) nodig. Om bij aanwezigheid van deze regel bovenstaande beslisbaarheidsredenering te kunnen handhaven, zouden we wensen dat deze regel alleen in de eerste stap van de afleiding kan worden toegepast. Dit kan bereikt worden door aan een grammatica G de volgende twee voorwaarden op te leggen:

1. Het startsymbool Z komt niet voor in de rechterleden van de regels;
2. alle regels, behalve eventueel (Z,Λ), zijn niet-dalend.

Een kleine modificatie van bovenstaande redenering leert, dat een grammatica die aan deze voorwaarden voldoet, beslisbaar is. In het bijzonder zijn grammatica's van type 1 beslisbaar: $L_1 \subset R$. Bij contekstvrije grammatica's komt voorwaarde 2. neer op: als $(A,\Lambda) \in R$, dan $A = Z$.

Onze methode is nu, dat we een gegeven contekstvrije grammatica G_0 (de "oude" grammatica) vervangen door een contekstvrije grammatica G_n (de "nieuwe" grammatica); de bedoeling daarbij is dat de beslisbaarheid van G_n gemakkelijker is vast te stellen dan die van G_0. Dit helpt alleen als ook uit de beslisbaarheid van G_n die van G_0 volgt. Om dit laatste te bewerkstelligen leggen we de volgende drie voorwaarden op aan de constructie van G_n uit G_0:

I. Als G_0 gegeven is, is de definitie van G_n (hulpsymbolen, startsymbool, regels) effectief uitvoerbaar;

II. $L(G_n) = L(G_0)$;

III. iedere afleiding(sboom) bij G_n kan effectief worden omgezet in een afleiding(sboom) bij G_0 met hetzelfde afgeleide woord.

Stel namelijk dat I, II, III vervuld zijn en G_n beslisbaar is. We kunnen bij gegeven G_0 de G_n maken op grond van I. Stel $x \in I_T^*$. De beslisbaarheid van G_n levert hetzij $x \notin L(G_n)$, maar dan geldt wegens II ook $x \notin L(G_0)$, hetzij een afleiding van x bij G_n, die op grond van III in een afleiding van x bij G_0 kan worden omgezet.

We maken nu eerst voorwaarde 1. in orde. Voor de definitie van G_n voegen we aan de hulpsymbolen van G_0 één nieuw hulpsymbool Z' toe, dat tevens startsymbool van G_n is. De regels van G_n zijn de regels van G_0 met daaraan toegevoegd (Z',Z). Het is gemakkelijk in te zien dat G_n een contekstvrije grammatica is, die aan voorwaarde 1. voldoet; bovendien is bij de constructie voldaan aan I, II, III.

We gaan nu een nieuwe grammatica definiëren die aan voorwaarde 2. voldoet en wel zo, dat als de oude grammatica aan voorwaarde 1. voldoet, dit ook geldt voor de nieuwe grammatica. Uiteraard moeten ook I, II en III vervuld zijn.

Een contekstvrije grammatica $G_0 = \langle I, I_T, Z, R \rangle$ is gegeven. Definieer $U := \{A \in I_N | A \overset{*}{\underset{G_0}{\Rightarrow}} \Lambda\}$. De grammatica G_n heeft dezelfde hulpsymbolen en hetzelfde startsymbool als G_0. De verzameling regels van G_n wordt, uitgaande van R, door de volgende voorschriften bepaald.

Laat alle (A,Λ) met $A \neq Z$ weg. Als $Z \in U$, voeg (zo nodig) (Z,Λ) toe. Voor iedere $(A,q) \in R$ met $q \neq \Lambda$ doen we het volgende.
Stel dat op m plaatsen in q een element van U staat. Als $q \notin U^*$, dan maak alle 2^m woorden r, die uit q ontstaan door in een willekeurige deelverzameling van de m plaatsen, waar elementen van U staan, deze elementen weg te laten en neem alle 2^m regels (A,r) in de verzameling regels van G_n op. Als $q \in U^*$ dan nemen we op analoge wijze de 2^m-1 woorden met uitzondering van het lege woord en daarvoor de regels (A,r). In beide gevallen is de regel (A,q) er zelf bij en hebben alle zo verkregen regels een niet-leeg rechterlid. We behandelen een voorbeeld. Stel $(A,A\alpha BAC) \in R$ met $A \in U$, $B \in U$, $C \notin U$, $\alpha \in I_T$. Nu is $m = 3$; we krijgen de 8 regels: $(A,A\alpha BAC)$, $(A,\alpha BAC)$, $(A,A\alpha AC)$, $(A,A\alpha BC)$, $(A,\alpha AC)$, $(A,\alpha BC)$, $(A,A\alpha C)$, $(A,\alpha C)$.

Een ander voorbeeld is $(A,ABA) \in R$, met $A \in U$, $B \in U$. Weer is $m = 3$; er komen nu 7 regels: (A,ABA), (A,BA), (A,AA), (A,AB), (A,A), (A,B), (A,A). We merken nog op dat er onder de verkregen regels gelijke kunnen zijn en dat het irrelevant is of het linkerlid van de regel al dan niet een element van U is.

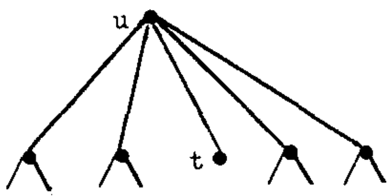
Hiermee is de definitie van G_n gegeven. Het is duidelijk, dat G_n contekstvrij is en aan de voorwaarde 2. voldoet. Eveneens duidelijk is dat als G_0 aan voorwaarde 1. voldoet dit ook voor G_n geldt. Wat voorwaarde I betreft, is het enige probleem om vast te stellen, dat U beslisbaar is: voor iedere $A \in I_N$ uitmaken of $A \in U$ dan wel $A \notin U$. Hiertoe stellen we het volgende vast:

$A \in U \Leftrightarrow$ er bestaat een A-boom bij G_0 met $f(t) \in I_N$ voor alle knopen $t \Leftrightarrow$
$\Leftrightarrow$ er bestaat een A-boom bij G_0 met $f(t) \in I_N$ voor alle knopen t en een minimaal aantal knopen.

Voor een boom van de laatste soort geldt dat de hoogte $\leq n$ (aantal hulpsymbolen). Het opschrijven van alle A-bomen van hoogte $\leq n$ bij een gegeven grammatica is effectief uitvoerbaar; men kan dan ook effectief vaststellen of daar een boom bij is, waar het lege woord als afgeleid woord bijhoort. Daarmee is vastgesteld of $A \in U$. Voor het vervolg merken we nog op, dat als $A \in U$, er tevens effectief een afleiding van Λ uit A is gevonden. Daarmee is I aangetoond.

We behandelen nu II en III tezamen. Stel eerst $x \in L(G_n)$. Dan is er een afleiding van x bij G_n. Beschouw een afleidingsstap, waarbij een regel is toegepast, die geen regel bij G_0 is. Dat kan (Z,Λ) zijn, maar dan is $Z \in U$ en kan de ene stap vervangen worden door een afleiding $Z \overset{*}{\underset{G_0}{\Rightarrow}} \Lambda$. De regel kan echter ook geweest zijn (A,r), waarbij r ontstaan is uit q door een aantal weglatingen van elementen van U en $(A,q) \in R$. In de oude grammatica passen we nu eerst (A,q) toe, gevolgd door een afleiding $X \overset{*}{\underset{G_0}{\Rightarrow}} \Lambda$ voor elke weggelaten X; dit resulteert samen in het effect van de toepassing van de regel (A,r). Op deze wijze kan de afleiding omgezet worden in een afleiding van x bij G_0; deze omzetting is bovendien effectief uitvoerbaar en levert $x \in L(G_0)$. We hebben daarmee III al aangetoond. Voor het bewijs van $L(G_0) \subset L(G_n)$ lassen we nog een intermediaire grammatica G_i in, die dezelfde hulpsymbolen en startsymbool heeft als G_0 en G_n. De regels zijn de regels van G_n met daaraan toegevoegd alle (A,Λ) met $A \in U$ en $A \neq Z$. Dan zijn ook alle regels van G_0 regels van G_i. Iedere afleidingsboom bij G_0 is dus ook afleidingsboom bij G_i en iedere afleidingsboom bij G_n is ook afleidingsboom bij G_i. In een afleidingsboom bij G_i noemen we een knoop t "verboden", als $f(t) \in I_N$ en de afleidingsregel die bij t wordt toegepast niet in G_n zit. Het is effectief vast te stellen of een knoop al dan niet verboden is. We beweren nu dat iedere afleidingsboom bij G_i effectief kan worden omgezet in een afleidingsboom bij G_n met hetzelfde afgeleide woord. Van deze bewering geven we slechts een schets van een bewijs. We passen volledige inductie toe naar het aantal knopen van de afleidingsboom. We onderscheiden twee gevallen:

1. De boom heeft geen verboden knopen. Dan is de boom zelf al een afleidingsboom bij G_n en zijn we klaar.
2. De boom heeft verboden knopen. Kies zo'n knoop t; dit is effectief uitvoerbaar. Stel $f(t) = A$, dan $A \neq Z$. Hieruit volgt, dat t niet de wortel



van de boom is en dus t een onmiddellijke voorganger u heeft. Omdat t een verboden knoop is zijn er geen knopen in de boom, die t als antecedent hebben. We proberen nu een nieuwe boom te maken door de knoop t eenvoudig weg te laten. Dit hoeft echter geen afleidingsboom op te leveren, omdat er een lacune valt: de knopen onder u "rechts" van t krijgen een verkeerde code en eveneens alle knopen die deze knopen als antecedent hebben. Er ontstaat wat we in §6 een quasi-afleidingsboom hebben genoemd; de codering daarvan kan worden gewijzigd zo, dat deze aan de eisen van een afleidingsboom voldoet. We gaan dat niet na. We moeten echter nog controleren of het ook met de toepassing van de regels in orde is. De enige knoop waarvoor dat niet vanzelf spreekt is de knoop u. Bij deze knoop was in de gegeven boom een regel met niet-leeg rechterlid toegepast, die dus een regel van G_n was; verder $A \in U$. Laten we nu uit het rechterlid van die regel dit voorkomen van A ook nog weg, dan krijgen we weer een regel van G_n en dus van G_i en die wordt in de nieuwe boom toegepast. Een uitzondering hierop krijgen we als de weglating van die A tot een leeg rechterlid zou leiden.

De situatie is dan zo, dat er vanuit u slechts één tak naar beneden gaat en wel die naar t. In dat geval heeft de toegepaste regel de gedaante (B,A) en is ontstaan uit een regel $(B,qAs) \in R$ met $q \in U^*$, $s \in U^*$. Aangezien ook $A \in U$, geldt dan ook $B \in U$. Maar dan is (B,A) een regel van G_i, want als $B = Z$, is (B,A) regel van G_n wegens $Z \in U$, en als $B \neq Z$, dan is ook (B,A) regel van G_i. Ook in dit geval is het dus in orde voor de knoop u. De nieuwe boom levert een afleidingsboom bij G_i met hetzelfde afgeleide woord en heeft één knoop minder. Volgens de inductieveronderstelling is deze boom effectief om te zetten in een afleidingsboom bij G_n met hetzelfde afgeleide woord. Dit voltooit het inductiebewijs.

Neemt men nu een $x \in L(G_0)$, dan bestaat er een afleidingsboom bij G_0 met afgeleid woord x. Deze is ook afleidingsboom bij G_i en dus om te zetten in een afleidingsboom bij G_n met afgeleid woord x. Dus $x \in L(G_n)$. Dit levert $L(G_0) \subset L(G_n)$ en daarmee is II aangetoond.

We hebben nog extra aangetoond dat een afleiding bij G_0 effectief kán worden omgezet in een afleiding bij G_n met hetzelfde afgeleide woord. Alles samen geeft dit de volgende stelling:

<u>Stelling 9.1.</u> Een contekstvrije grammatica is beslisbaar.

In de notatie van §3 is dit te schrijven als $\mathcal{L}_2 \subset \mathcal{R}$. We hebben trouwens ook bewezen dat $\mathcal{L}_2 \subset \mathcal{L}_1$.

We noemen nog een aantal onbeslisbaarheidsresultaten, veelal zonder bewijs. Het is onbeslisbaar of de talen gegenereerd door twee contekstvrije grammatica's aan elkaar gelijk zijn. Dit moet als volgt worden opgevat. Men kan zonder moeite paren grammatica's maken, waarvan men vast kan stellen dat ze **niet** dezelfde talen genereren. Wat echter onmogelijk is, is het maken van een effectieve procedure, waar men twee willekeurige contekstvrije grammatica's kan instoppen en die als antwoord levert of de bijbehorende talen al dan niet gelijk zijn. Eveneens onbeslisbaar is of de taal van een contekstvrije grammatica deelverzameling is van de taal van een andere contekstvrije grammatica.

Wel beslisbaar is of de taal van een contekstvrije grammatica leeg is en ook of deze eindig is. We geven een schets van het bewijs van deze beweringen. Uiteraard geldt:

> $L(G) \neq \emptyset \Leftrightarrow$ er bestaat een afleidingsboom bij G $\Leftrightarrow$ er bestaat een afleidingsboom bij G met een minimaal aantal knopen.

Voor zo'n boom geldt echter weer, dat de hoogte $\leq$ aantal hulpsymbolen. Of zulk een afleidingsboom bestaat kan effectief worden vastgesteld. Voor de eindigheid bedenken we, dat we bij het bewijs van Stelling 8.4 bij een contekstvrije grammatica G effectief een getal p hebben gevonden zo, dat als er een $w \in L(G)$ bestaat met $\ell(w) > p$, dan $L(G)$ oneindig. We kunnen nu ook bij een gegeven contekstvrije grammatica G en een getal p effectief een contekstvrije grammatica maken die de taal $L(G) \cap \{x \in I_T^* | \ell(x) > p\}$ genereert. De beslisbaarheid dat $L(G)$ eindig is volgt dan uit

$$L(G) \text{ eindig} \Leftrightarrow L(G) \cap \{x \in I_T^* | \ell(x) > p\} = \emptyset .$$

Het is onbeslisbaar of een contekstvrije grammatica een reguliere taal genereert. Het is ook onbeslisbaar of het complement van een door een contekstvrije grammatica gegenereerde taal leeg is, of eindig is, of contekstvrij is. Onbeslisbaar is ook of de doorsnede van de door twee contekstvrije grammatica's gegenereerde talen leeg is, of contekstvrij is.

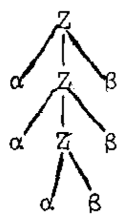
Voor 3-type talen is veel meer beslisbaar. Een 3-type grammatica als generator en een eindige automaat als acceptor van een zelfde taal zijn effectief in elkaar om te zetten. Verder kan men voor door eindige automaten geaccepteerde talen ook effectief automaten maken voor vereniging, doorsnede en complement van talen. Dit zelfde geldt dus ook voor 3-type grammatica's. Bedenk verder dat $L = M \Leftrightarrow (L \cap (I_T^* \backslash M)) \cup (M \cap (I_T^* \backslash L)) = \emptyset$. Als 3-type grammatica's voor L en M gegeven zijn, dan is effectief een 3-type grammatica voor $(L \cap (I_T^* \backslash M)) \cup (M \cap (I_T^* \backslash L))$ te maken en is het dus beslisbaar of deze taal leeg is. Het is dus ook beslisbaar of $L = M$. Het geval $L \subset M$ behandelt men op analoge wijze met behulp van $L \cap (I_T^* \backslash M) = \emptyset$.

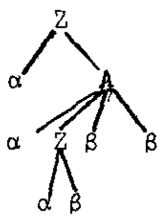
Definitie ambigue grammatica

Een contekstvrije grammatica G heet ambigu als er twee verschillende afleidingsbomen bij G bestaan met hetzelfde afgeleide woord.

Neem als voorbeeld $I_N = \{Z, A\}$, $I_T = \{\alpha, \beta\}$,

$R = \{(Z, \alpha Z \beta), (Z, \alpha A), (Z, \alpha\beta), (A, \alpha Z \beta\beta)\}$. De volgende twee bomen





zijn afleidingsbomen bij deze grammatica, die hetzelfde afgeleide woord $\alpha\alpha\alpha\beta\beta\beta$ hebben. De grammatica is dus ambigu. Het is gemakkelijk na te gaan, dat de door deze grammatica gegenereerde taal $\{\alpha^{n+1}\beta^{n+1} \mid n \in \mathbb{N}\}$ is. Deze taal wordt echter ook gegenereerd door de volgende contekstvrije grammatica: $I_N = \{Z\}$, $I_T = \{\alpha, \beta\}$, $R = \{(Z, \alpha Z \beta), (Z, \alpha\beta)\}$, waarvan eenvoudig in te zien is dat zij niet ambigu is.

In de meeste toepassingen zijn ambigue grammatica's ongewenst. Men wil dus een contekstvrije taal graag genereren met een niet-ambigue contekstvrije grammatica. Dit kan echter niet altijd.

Definitie inherent ambigue contekstvrije taal

Een contekstvrije taal heet inherent ambigu als iedere contekstvrije grammatica die deze taal genereert ambigu is.

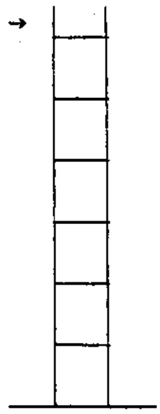
De taal $\{\alpha^n\beta^n\gamma^m | n \in \mathbb{N}, m \in \mathbb{N}\} \cup \{\alpha^m\beta^n\gamma^n | n \in \mathbb{N}, m \in \mathbb{N}\}$ is inherent ambigu. Dit bewijzen we niet.

Het is van praktisch belang om van een contekstvrije grammatica vast te stellen of zij ambigu is en van een contekstvrije taal of hij inherent ambigu is. Een algemene methode daarvoor bestaat echter niet op grond van het volgende.

Het is onbeslisbaar of een contekstvrije grammatica ambigu is en ook of een contekstvrije taal inherent ambigu is.

§10. STAPELAUTOMATEN

We bespreken nu een uitbreiding van de eindige automaat, die geschikt is om contekstvrije talen te accepteren. Om een echte uitbreiding te krijgen zal zo'n automaat een geheugen moeten hebben, waarin een onbegrensde hoeveelheid informatie kan worden opgeslagen. Dit betekent, dat deze hoeveelheid op ieder moment eindig is, maar dat er voor deze hoeveelheid geen bovengrens is. Dit wordt gerealiseerd met een stapel. Deze wordt fysisch beschreven met een verticaal geplaatste band, die in hokjes verdeeld is.



In elk hokje staat een symbool geschreven. De stapel is alleen aan de bovenzijde toegankelijk. Daar kan een symbool worden gelezen en kan worden uitgeveegd of bijgevoegd; daarbij kan de hoogte van de stapel veranderen. Formeel doen we dit door het bovenste symbool ξ te lezen en te verwijderen en te vervangen door een woord w. Als $w = \Lambda$, dan is het symbool uitgeveegd; als $w = \xi$, dan is er niets veranderd; als $w = \xi y$ dan is er iets toegevoegd; als $w = \eta y$ met $\eta \neq \xi$, dan is ξ door η vervangen en y toegevoegd. Voor het overige verkeert de automaat in een toestand, leest een invoersymbool en het bovenste symbool van de stapel en gaat vervolgens over in een nieuwe toestand en schrijft op de stapel. We laten nu echter ook nog toe, dat de automaat een handeling verricht zonder een invoersymbool te lezen; hij verandert dan op grond van de geldende toestand en het bovenste stapelsymbool zijn toestand en schrijft op de stapel. Hiermee hebben we een niet-deterministisch gedrag geïntroduceerd: de automaat heeft keuze tussen het doen van een stap met lezen of zonder lezen van de invoer. We breiden dit nu uit door in elke situatie zowel bij lezen als bij niet-lezen meer dan één keuze toe te staan, waarbij we echter wel in elke situatie het aantal mogelijke keuzen eindig willen houden. Deze eindigheid is niet automatisch vervuld, omdat er voor het op de stapel te schrijven woord w oneindig veel mogelijkheden zijn. Formeel wordt dit beschreven door als waarde van een afbeeldingsfunctie een eindige deelverzameling van de verzameling der mogelijke uitkomsten te nemen. We gebruiken daarbij de notatie, dat als V een verzameling is, $P_\omega(V)$ de verzameling is van alle eindige deelverzamelingen van V. Dat een actie van de automaat twee uitkomsten geeft, namelijk een toestand en een woord op de stapel vangen we op met een Cartesisch produkt. Een stap waarbij niet gelezen wordt, wordt formeel beschreven met het lezen van Λ. Voor de symbolen op de stapel kiezen we een aparte verzameling symbolen, die overigens niet hoeft te verschillen van de verzameling invoersymbolen. Omdat de automaat zijn werking verricht mede op grond van het bovenste symbool op de stapel, zal er ook bij het begin van de werking iets op de stapel moeten staan; daartoe kiezen we een beginsymbool voor de stapel. Verder kan de hierboven genoemde eindige verzameling uitkomsten ook leeg zijn; als dat zo is, is de werking van de automaat geblokkeerd. Blokkade zal ook optreden als tijdens de werking van de automaat de stapel leeg raakt. De inhoud van de stapel zullen we formeel beschrijven met een symboolrij, die echter horizontaal is en de stapel verticaal. Voor deze overgang moeten we een afspraak maken; we stellen vast dat de bovenkant van de stapel overeenkomt met de linkerkant van de symboolrij.

Een automaat, die met bovenstaande beschrijving correspondeert, wordt een stapelautomaat genoemd. De Engelse benaming is pushdown automaton. Deze naam correspondeert met het beeld, dat de stapel niet aan de onderzijde, maar aan de bovenzijde wordt vastgehouden. Bij toevoeging van symbolen moet de stapel naar beneden worden gedrukt, bij verwijdering van symbolen moet de stapel uit zichzelf omhoog komen.

Definitie stapelautomaat

Een 7-tupel $A = \langle S,I,\Gamma,\delta,s_0,Z_0,F\rangle$ is een stapelautomaat, als geldt:

S is een eindige verzameling (elementen van S heten toestanden),

I is een eindige niet-lege verzameling (elementen van I heten invoersymbolen),

Γ is een eindige verzameling (elementen van Γ heten stapelsymbolen),

δ is een afbeelding $S \times (I \cup \{\Lambda\}) \times \Gamma \to P_\omega(S \times \Gamma^*)$,

$s_0 \in S$ (begintoestand),

$Z_0 \in \Gamma$ (beginsymbool),

$F \subset S$ (elementen van F heten toegelaten toestanden).

De interpretatie van δ is, dat als $(s',w) \in \delta(s,\xi,\eta)$ en $\xi \in I$, de automaat, als hij in toestand s verkeert en η op de top van de stapel heeft staan, bij lezing van het invoersymbool ξ in één stap over kan gaan in toestand s' onder gelijktijdige vervanging van η bovenaan de stapel door w. Een analoge interpretatie, maar dan zonder lezing van een invoersymbool, geven we als in het bovenstaande ξ door Λ wordt vervangen.

Om nu de verwerking van een invoerwoord door de automaat te beschrijven voeren we in een bij de automaat behorende <u>momentane beschrijving</u>, dat is een tripel (s,p,q) waarin $s \in S$, $p \in I^*$, $q \in \Gamma^*$. De interpretatie hiervan is, dat s de geldende toestand is, p het nog niet gelezen deel van het invoerwoord, q de geldende inhoud van de stapel. We definiëren nu een binaire relatie $\vdash_{\overline{A}}$ tussen momentane beschrijvingen. We willen dat zo doen, dat $m \vdash_{\overline{A}} m'$ geïnterpreteerd kan worden als: m kan in één berekeningsstap bij A overgaan in m'. We definiëren dat

$$(s,\xi p,Tq) \vdash_{\overline{A}} (s',p,q'q) \quad \text{geldt}$$

voor $s \in S$, $\xi \in I \cup \{\Lambda\}$, $p \in I^*$, $T \in \Gamma$, $q \in \Gamma^*$, als $(s',q') \in \delta(s,\xi,T)$.

We definiëren $\vdash^{*}_{\overline{A}}$ als de reflexief-transitieve afsluiting van $\vdash_{\overline{A}}$:
$m \vdash^{*}_{\overline{A}} m'$ geldt per definitie als er een $n \in \mathbb{N}$ en een rij $m_0,\ldots,m_n$ van momentane beschrijvingen bestaan zo, dat $m_0 = m$, $m_n = m'$ en $m_j \vdash_{\overline{A}} m_{j+1}$ voor $j \in \{0,\ldots,n-1\}$.

We willen de automaat als acceptor gebruiken. Dit kan echter op verschillende manieren; we beschrijven de twee meest gebruikelijke keuzen. De eerste is om evenals bij de eindige automaat gebruik te maken van de verzameling F van toegelaten toestanden. Op grond van het non-determinisme beschrijven we het zo, dat we een woord w accepteren, als er een rij overgangen bestaat, die begint in de begintoestand en met Z_0 op de stapel, waarbij het hele woord w gelezen wordt en geëindigd in een toestand in F. In plaats van aldus op grond van toegelaten toestanden te accepteren, doet men dit ook wel op grond van een lege stapel: met een soortgelijke start als zoëven moet het mogelijk zijn om na lezing van het gehele woord te eindigen met een lege stapel. Formeel leidt dit tot de volgende twee definities voor een stapelautomaat A:

$$L_1(A) := \{x \in I^* \mid \exists_{q \in \Gamma^*} \exists_{s \in F} [(s_0, x, Z_0) \overset{*}{\underset{A}{\vdash}} (s, \Lambda, q)]\} .$$

$$L_2(A) := \{x \in I^* \mid \exists_{s \in S} [(s_0, x, Z_0) \overset{*}{\underset{A}{\vdash}} (s, \Lambda, \Lambda)]\}.$$

Bij $L_2(A)$ speelt de verzameling F in het geheel geen rol. Als afkorting van een stapelautomaat wordt soms ndpa gebruikt (non-deterministic pushdown automaton).

Het verband tussen $L_1(A)$ en $L_2(A)$ komt tot uitdrukking in de volgende stelling:

<u>Stelling 10.1</u>. Bij iedere ndpa A bestaan er ndpa's A' en A" zo, dat $L_2(A') = L_1(A)$ en $L_1(A'') = L_2(A)$.

<u>Bewijs</u>: We geven alleen de definities van A' en A" en laten het bewijs dat $L_2(A') = L_1(A)$ en $L_1(A'') = L_2(A)$ aan de lezer over.

Stel $A = \langle S,I,\Gamma,\delta,s_0,Z_0,F\rangle$. Kies een nieuwe toestand s' en definieer $S' := S \cup \{s'\}$; kies twee nieuwe stapelsymbolen Z_0' en Z' en definieer $\Gamma' := \Gamma \cup \{Z_0', Z'\}$. Verder $s_0' := s_0, Z_0' := Z_0'$, $F' = \emptyset$. De afbeelding δ' wordt als volgt gedefinieerd. Stel $\xi \in I \cup \{\Lambda\}$, $s \in S$, $t \in S'$, $T \in \Gamma$:

$$\delta'(t,\xi,Z_0') := \begin{cases} \{(s_0, Z_0 Z')\} & \text{als } t = s_0 \text{ en } \xi = \Lambda, \\ \emptyset & \text{als } t \neq s_0 \text{ of } \xi \neq \Lambda, \end{cases}$$

$$\delta'(s,\xi,T) := \begin{cases} \delta(s\ ,\ \xi,T) & \text{als } s \notin F \text{ of } \xi \neq \Lambda, \\ \delta(s,\xi,T) \cup \{(s',\Lambda)\} & \text{als } s \in F \text{ en } \xi = \Lambda, \end{cases}$$

$$\delta'(s',\xi,T) := \begin{cases} \{(s',\Lambda)\} & \text{als } \xi = \Lambda\ , \\ \emptyset & \text{als } \xi \neq \Lambda, \end{cases}$$

$$\delta'(t,\xi,Z') := \begin{cases} \{(s',\Lambda)\} & \text{als } t \in F \cup \{s'\} \text{ en } \xi = \Lambda, \\ \emptyset & \text{als } t \notin F \cup \{s'\} \text{ of } \xi \neq \Lambda. \end{cases}$$

Daarmee is de definitie van A' voltooid. Kies nu een nieuwe toestand s'' en definieer $S'' := S \cup \{s''\}$; kies twee nieuwe stapelsymbolen Z_0'' en Z'' en definieer $\Gamma'' := \Gamma \cup \{Z_0'', Z''\}$. Verder $s_0'' := s_0$, $Z_0'' := Z_0''$, $F'' := \{s''\}$. De afbeelding δ'' wordt als volgt gedefinieerd. Stel $\xi \in I \cup \{\Lambda\}$, $s \in S''$, $T \in \Gamma$:

$$\delta''(s,\xi,Z_0'') := \begin{cases} \{(s_0, Z_0 Z'')\} & \text{als } s = s_0 \text{ en } \xi = \Lambda, \\ \emptyset & \text{als } s \neq s_0 \text{ of } \xi \neq \Lambda, \end{cases}$$

$$\delta''(s,\xi,T) := \begin{cases} \delta(s,\xi,T) & \text{als } s \neq s'', \\ \emptyset & \text{als } s = s'', \end{cases}$$

$$\delta''(s,\xi,Z'') := \begin{cases} \{(s'',Z'')\} & \text{als } s \neq s'' \text{ en } \xi = \Lambda, \\ \emptyset & \text{als } s = s'' \text{ of } \xi = \Lambda. \end{cases}$$

Daarmee is de definitie van A'' voltooid. ∎

De nu volgende stelling geeft het verband met contekstvrije ralen.

Stelling 10.2 (Chomsky). Een taal L is contekstvrij dan en slechts dan als er een ndpa A bestaat zo, dat $L = L_2(A)$.

Bewijs: Stel L contekstvrij. Kies een contekstvrije grammatica $G = \langle I_N \cup I_T, I_T, Z, R \rangle$ zo, dat $L = L(G)$. We definiëren een ndpa A als volgt. $S := \{0\}$, $I := I_T$, $\Gamma := I_N \cup I_T$, $s_0 := 0$, $Z_0 := Z$, $F := \emptyset$. De afbeelding δ wordt als volgt gedefinieerd. Stel $\xi \in I_T \cup \{\Lambda\}$, $\eta \in I_T$, $A \in I_N$:

$$\delta(0,\xi,\eta) := \begin{cases} \{(0,\Lambda)\} & \text{als } \xi = \eta, \\ \emptyset & \text{als } \xi \neq \eta, \end{cases}$$

$$\delta(0,\xi,A) := \begin{cases} \{(0,r) \mid (A,r) \in R\} & \text{als } \xi = \Lambda, \\ \emptyset & \text{als } \xi \neq \Lambda. \end{cases}$$

Daarmee is de definitie van A voltooid. Het bewijs dat $L_2(A) = L(G)$ laten we aan de lezer over.

Omgekeerd gaan we nu uit van een ndpa $A = \langle S, I, \Gamma, \delta, s_0, Z_0, F \rangle$. We definieren een contekstvrije grammatica als volgt: $I_N := S \times S \times \Gamma$, $I_T := I$,

$Z := (s_0, s_0, Z_0)$,

$$R := \{((s,t_m,T), \xi(t_0,t_1,T_0)\ldots(t_{m-1},t_m,T_{m-1})) \mid m \in \mathbb{N}, t_0 \in S, \ldots, t_m \in S, s \in S, T_0 \in \Gamma, \ldots, T_{m-1} \in \Gamma, T \in \Gamma, \xi \in I \cup \{\Lambda\}, (t_0, T_0 \ldots T_{m-1}) \in \delta(s,\xi,T)\}.$$

Let wel, dat dit voor $m = 0$ een regel $((s,t_0,T),\xi)$ oplevert onder de voorwaarde $(t_0,\Lambda) \in \delta(s,\xi,T)$. Daarmee is de definitie van G voltooid. Het bewijs dat $L(G) = L_2(A)$ laten we aan de lezer over. ∎

Op grond van Stelling 10.1 is er ook een met Stelling 10.2 corresponderende stelling, waarin L_2 door L_1 is vervangen.

Bij het nabootsen van een automaat door een fysische machine is het indeterminisme een hinderlijke omstandigheid. We vragen ons af of wij ons tot deterministische automaten hadden kunnen beperken. We moeten die dan eerst definiëren. De bedoeling is, dat de automaat in iedere situatie niet meer dan één keuze heeft; daarbij inbegrepen de keuze tussen wel en niet lezen. We laten echter wel blokkades toe, dat wil zeggen situaties waarin er in het geheel geen keuze is.

Definitie deterministische stapelautomaat

Een ndpa $\langle S,I,\Gamma,\delta,s_0,Z_0,F\rangle$ heet een deterministische stapelautomaat als geldt:

$$\forall_{s\in S}\ \forall_{\xi\in I}\ \forall_{T\in\Gamma}[\#(\delta(s,\xi,T)) + \#(\delta(s,\Lambda,T)) \leq 1] .$$

Dit betekent dat deze aantallen 0,0 of 1,0 of 0,1 zijn. We gebruiken de afkorting dpa voor een deterministische stapelautomaat. Voor deze automaten maakt het wel verschil of er op grond van toegelaten toestanden, dan wel op grond van lege stapel wordt geaccepteerd. Enerzijds is het wel zo, dat er bij iedere dpa A een dpa A" bestaat zo, dat $L_1(A'') = L_2(A)$, maar anderzijds bestaat er een dpa A zo, dat voor alle dpa's A' geldt $L_2(A') \neq$ $\neq L_1(A)$. Acceptatie op toegelaten toestanden levert het meest aanvaardbare resultaat.

Definitie deterministische taal

Een taal L heet deterministisch als er een dpa A bestaat zo, dat $L = L_1(A)$.

Uiteraard is een deterministische taal contekstvrij. Dat niet iedere contekstvrije taal deterministisch is volgt uit de volgende twee eigenschappen van deterministische talen, die we niet zullen bewijzen.

<u>Stelling 10.3</u>. Het complement van een deterministische taal is deterministisch.

<u>Stelling 10.4</u>. Als L een deterministische taal is dan bestaat er een niet-ambigue contekstvrije grammatica G zo, dat L = L(G).

Een contekstvrije taal, waarvan het complement niet contekstvrij is, is op grond van Stelling 10.3 niet deterministisch en zulke talen bestaan. Een inherent ambigue contekstvrije taal is op grond van Stelling 10.4 niet deterministisch en zulke talen bestaan.

Het omgekeerde van Stelling 10.4 is onjuist. De taal van de palindromen van even lengte (zie §1) kan worden gegenereerd met een niet-ambigue contekstvrije grammatica, maar is niet deterministisch. Dit tonen we niet aan.

<u>Opgave</u>. Construeer een ndpa die de taal der palindromen van even lengte accepteert.

We beschouwen nu de stappen van een automaat, waarbij geen invoersymbool wordt gelezen (Λ-stappen) en vragen ons af of deze gemist kunnen worden. Dit zou een vereenvoudiging betekenen, omdat dan bij verwerking van een invoerwoord het aantal stappen gelijk is aan de lengte van dat woord. Formeel definiëren we een stapelautomaat zonder Λ-stappen als een stapelautomaat $\langle S.I,\Gamma,\delta,s_0,Z_0,F\rangle$, waarvoor geldt: $\forall_{s\in S}\ \forall_{T\in\Gamma}\ [\delta(s,\Lambda,T) = \emptyset]$.

Het is duidelijk, dat voor een stapelautomaat A zonder Λ-stappen geldt $\Lambda \notin L_2(A)$. Dit blijkt de enige beperking te zijn.

<u>Stelling 10.5</u>. Als A een ndpa is, dan bestaat er een ndpa A' zonder Λ-stappen zo, dat $L_1(A') = L_1(A)$. Als bovendien geldt $\Lambda \notin L_2(A)$, dan bestaat er een ndpa A" zonder Λ-stappen zo, dat $L_2(A") = L_2(A)$.

Deze stelling bewijzen we niet. Er blijkt uit, dat we ons voor het accepteren van contekstvrije talen tot stapelautomaten zonder Λ-stappen hadden kunnen beperken. Dat neemt niet weg, dat het voor een gegeven contekstvrije taal soms gemakkelijker is om een stapelautomaat met Λ-stappen te construeren, die de taal accepteert, dan een zonder Λ-stappen. Bovendien levert de combinatie van determinisme en het weglaten van Λ-stappen wel een essentiële beperking op. Er bestaan namelijk deterministische talen, die niet door een dpa zonder Λ-stappen kunnen worden geaccepteerd. Voor zulke talen bestaat er dan wel een dpa met Λ-stappen of een ndpa zonder Λ-stappen. Voor de definitie van deterministische talen blijkt het daarom gewenst om stapelautomaten met Λ-stappen mee in de beschouwing te betrekken.

HOOFDSTUK 3. SEMANTIEK

§11. INLEIDING

Het opstellen van een formalisme geschiedt meestal met een zekere bedoeling. Zeer algemeen geformuleerd houdt semantiek zich bezig met het verband tussen formele systemen en datgene wat ermee bedoeld is (de "betekenis"). Als men bijvoorbeeld een contekstvrije grammatica opstelt ten behoeve van natuurlijke talen of programmeertalen, dan kiest men bij voorkeur hulpsymbolen, die iets betekenen en waarvoor men ook namen kiest, waaruit die betekenis blijkt (voorbeelden: (noun phrase), (verb phrase) bij natuurlijke talen; <identifier>, <integer>, <procedure> bij programmeertalen). Het opstellen van een afleidingsboom voor een tot de taal behorende "tekst" zal dan een bijdrage leveren tot het vinden van de "betekenis" van die tekst. Op de weg naar de betekenis is dat echter pas een eerste stap, die trouwens gewoonlijk nog niet tot de semantiek gerekend wordt. Toch houdt bij voorbeeld ambiguïteit al in dat er meer dan één interpretatie van de tekst mogelijk is. Wil men verder gaan, dan zal men zijn doelstellingen nader moeten preciseren.

Het voorbeeld van een programmeertaal is illustratief. Een programmeertekst zal vaak geschreven worden met het oogmerk deze vervolgens door een rekenautomaat te laten lezen en uitvoeren. Men zou dan onder semantiek kunnen verstaan de betrekking tussen de gedaante van de tekst en de serie handelingen, die in de automaat worden verricht, of eventueel alleen de betrekking tussen eindtoestand en begintoestand van de automaat. Het feit dat een zelfde tekst door automaten van verschillend type en merk kan worden uitgevoerd , zouden we kunnen ondervangen door de introductie van een "abstracte" automaat, waarvan verondersteld wordt dat zijn gedrag representatief is voor het gedrag van de "fysische" automaten. Een schrijver van een programmatekst zal echter in het algemeen verderstrekkende oogmerken hebben dan alleen het bezighouden van een rekenautomaat. Hoewel deze oogmerken zeer divers kunnen zijn, beschrijven we ze hier als "het oplossen van een probleem". Het oplossen van de vraag of men die doelstelling heeft verwezenlijkt is pas de ware semantiek; de beperking tot het "computergedrag" wordt soms wel pragmatiek of operationele semantiek genoemd. Het ontwerpen van mathematische modellen voor de semantiek in wijdere zin noemt men mathematische semantiek.

Dat een programma, dat pretendeert de grootste gemene deler van twee positieve natuurlijke getallen uit te rekenen, dat oók werkelijk doet, is alleen vast te stellen, als men weet wat de grootste gemene deler van twee positieve natuurlijke getallen is. Als men zich hier van af maakt door te stellen dat de grootste gemene deler gedefinieerd is door hetgeen het programma oplevert, gebruikt men de term "grootste gemene deler" in een andere betekenis, dan gebruikelijk is; dit geldt ook als het programma correct is.

Het is niet de bedoeling om hier een systematische behandeling te geven van de semantiek van programmeertalen. We volstaan met het belichten van enkele mathematische achtergronden.

Een uitvoering van een programma leidt tot vergroting van informatie. Als het programma bijvoorbeeld de waarde van een funtie f uitrekent, dan levert de invoer van een argumentwaarde a als uitvoer $f(a)$. Dit vergroten van de informatie komt in het bijzonder tot uiting als de functie herhaaldelijk wordt uitgerekend zoals dat het geval is bij de uitvoering van een recursieve procedure. Veronderstel dat gevraagd is om n! uit te rekenen en dat vermenigvuldiging tot de normale rekenfaciliteiten behoort, dan kan het doel bereikt worden met een recursieve procedure fac waarbij fac(n) gedefinieerd is door de programmatekst:

$$\underline{\text{if}}\ n = 0\ \underline{\text{then}}\ 1\ \underline{\text{else}}\ n * \text{fac}(n-1)\ .$$

Ter verklaring vatten we de tekst van de procedure eerst op als een operator P, die werkt op een variabele functie f en waarbij (Pf)(n) gedefinieerd wordt door

$$\underline{\text{if}}\ n = 0\ \underline{\text{then}}\ 1\ \underline{\text{else}}\ n * f(n-1)\ .$$

We veronderstellen dat de variabele n de verzameling $\mathbb{Z}$ der gehele getallen doorloopt. Voor iedere functie f is Pf ook een functie, dus P is een operator die functies in functies omzet. Kiest men bijvoorbeeld voor f de identieke functie dan is $(Pf)(0) = 1$ en $(Pf)(n) = n(n-1)$ als $n \neq 0$. Nu is fac blijkbaar een functie, die voldoet aan Pfac = fac, maar fac(n) is alleen gedefinieerd voor $n \geqslant 0$, en de gebruikelijke wijze van uitvoering van bovenstaande programmatekst voor fac met een negatief getal als invoer leidt tot een niet-stoppende berekening. Dat neemt niet weg, dat Pf = f wel degelijk oplossingen heeft, die voor alle $n \in \mathbb{Z}$ gedefinieerd zijn. Zo voldoet voor iedere constante c de functie g_c gedefinieerd door

$$g_c(n) := \begin{cases} n! & \text{als } n \geqslant 0\ , \\ \dfrac{(-1)^{n+1}c}{(-n-1)!} & \text{als } n < 0\ , \end{cases}$$

aan $Pg_c = g_c$. Bij de gebruikelijke uitvoering van de recursieve procedure

is het zo, dat naarmate het aantal malen dat de procedure mag worden aangeroepen groter is, het aantal waarden van n waarvoor fac(n) wordt verkregen, ook groter is. We kunnen dit beschrijven met behulp van partiële functies, dat zijn functies f die op een deelverzameling van $\mathbb{Z}$ gedefinieerd zijn. De groei van die deelverzameling weerspiegelt de groei van onze kennis omtrent de te berekenen functie.
Als we aannemen, dat we met totale onwetendheid beginnen, dan stellen we dat f_0 de nergens gedefinieerde functie is. Met behulp van de operator P maken we dan de rij functies $f_0, f_1, \ldots,$ die inductief gedefinieerd wordt door $f_{n+1} = Pf_n$. Bij fac is het duidelijk dat de uitkomst wordt, dat $f_n(x) = x!$ voor $0 \leqslant x < n$ en $f_n(x)$ ongedefinieerd voor alle andere $x \in \mathbb{Z}$. Bij uitvoering is f_n de functie die verkregen wordt als men de procedure ten hoogste n maal aanroept. Elke volgende functie is een voortzetting van de vorige. De door de recursieve procedure bepaalde functie f is de omvattende functie $f(x) = x!$ als $x \geqslant 0$, $f(x)$ ongedefinieerd als $x < 0$. We streven nu een iets algemenere formulering na.

§12. PARTIËLE FUNCTIES

Stel A en B verzamelingen en T de verzameling van alle partiële functies van A naar B, dat wil zeggen bij iedere $f \in T$ behoort een domein $D_f \subset A$ en voor alle $x \in D_f$ geldt $f(x) \in B$. Op T definiëren we een relatie $\leqslant$ door voor $f \in T$ en $g \in T$ te stellen

$$f \leqslant g :\Leftrightarrow D_f \subset D_g \text{ en } \forall_{x \in D_f} [f(x) = g(x)] .$$

In woorden zegt men: "g is voortzetting van f" of "g geeft meer informatie dan f". De relatie $\leqslant$ is wiskundig eenvoudig, als we de verzamelingstheoretische opvatting kiezen, waarbij functies verzamelingen van paren zijn:

$f = \{(x,f(x)) | x \in D_f\}$. Bij die opvatting is $f \leqslant g$ hetzelfde als $f \subset g$. Daaruit volgt direct dat $\leqslant$ aan de eisen van een partiële ordening voldoet:

$$f \leqslant f\ ,$$

$$\text{als } f \leqslant g \text{ en } g \leqslant h, \text{ dan } f \leqslant h,$$

$$\text{als } f \leqslant g \text{ en } g \leqslant f, \text{ dan } f = g\ .$$

We beschouwen nu de operatoren $P: T \to T$. We zullen aan de operatoren enkele voorwaarden opleggen, die geïnspireerd zijn door de uitvoering van programmateksten. We eisen eerst, dat als g meer informatie bevat dan f, ook Pg meer informatie bevat dan Pf. Dit leidt tot de volgende voorwaarde:

1. Als $f \leqslant g$, dan $Pf \leqslant Pg$ (monotonie).

De tweede voorwaarde komt voort uit de gedachte, dat als de berekening van $(Pf)(x)$ een uitkomst oplevert, er bij de uitvoering hiervan de berekening van $f(y)$ voor slechts eindig veel waarden van y wordt gevergd. Welke waarden van y dat zijn mag daarbij behalve van f ook van de waarde van x afhangen. Vervangen we nu echter de f door een g, die in alle y, waarvoor de $f(y)$ bij de berekening van $(Pf)(x)$ gebruikt worden, dezelfde waarde heeft als f, dan verloopt de berekening geheel op dezelfde wijze en is $(Pg)(x) = (Pf)(x)$.

Dit leidt tot de volgende voorwaarde:

2. Als $f \in T$ en $x \in D_{Pf}$, dan bestaat er een eindige deelverzameling Y van D_f zo, dat voor alle $g \in T$, waarvoor geldt $Y \subset D_g$ en $\forall_{y \in Y}[g(y) = f(y)]$, ook geldt $x \in D_{Pg}$ en $(Pg)(x) = (Pf)(x)$.

We kunnen een andere formulering van 2. krijgen door in plaats van de verzameling Y de restrictie van f tot Y te gebruiken. We noemen een $h \in T$ eindig als D_h eindig is (de verzameling h is dan ook eindig). We definiëren T_0 als de verzameling van de eindige elementen van T.

2'. Als $f \in T$ en $x \in D_{Pf}$, dan bestaat er een $h \in T_0$ zo, dat $h \leqslant f$ en voor alle $g \in T$, waarvoor geldt $h \leqslant g$, ook geldt $x \in D_{Pg}$ en $(Pg)(x) = (Pf)(x)$.

Het is duidelijk dat 2. en 2'. equivalent zijn. Als we aannemen dat P ook monotoon is, kan 2. in een eenvoudiger gedaante gebracht worden.

2". Als $f \in T$ en $x \in D_{Pf}$, dan bestaat er een $h \in T_0$ zo, dat $h \leqslant f$ en $x \in D_{Ph}$.

Het is duidelijk dat 2'. $\Rightarrow$ 2". Dat uit 1. en 2". volgt dat 2'. geldt ziet men als volgt. Stel $f \in T$ en $x \in D_{Pf}$ dan geeft 2". dat er een $h \in T_0$ is met $h \leqslant f$ en $x \in D_{Ph}$. Als nu voor $g \in T$ geldt $h \leqslant g$, dan volgt uit 1. dat $Ph \leqslant Pg$, dus $x \in D_{Pg}$ en $(Pg)(x) = (Ph)(x) = (Pf)(x)$, waarmee 2'. is aangetoond.

We veronderstellen nu dat de operator $P: T \to T$ aan 1. en 2. voldoet. We definiëren een rij $f_0, f_1, \ldots$ van elementen van T met volledige inductie: $f_0 = \emptyset$ (nergens gedefinieerde functie) en $f_{n+1} = Pf_n$ (we schrijven dan ook $f_n = P^n f_0$). Er geldt nu $\forall_{n \in \mathbb{N}} [f_n \leqslant f_{n+1}]$; de rij is monotoon niet-dalend. Dit bewijzen we met volledige inductie. $f_0 = \emptyset \leqslant f_1$ (triviaal). Stel $f_n \leqslant f_{n+1}$, dan $f_{n+1} = Pf_n \leqslant Pf_{n+1} = f_{n+2}$ op grond van 1. Met de verzamelingstheoretische opvatting van functies definiëren we nu $f := \bigcup \{f_n \mid n \in \mathbb{N}\}$. We bewijzen dat dit een partiële functie is. Het is uiteraard een verzameling paren. Stel nu $(x,y) \in f$ en $(x,z) \in f$. Dan is er een $n \in \mathbb{N}$ zo, dat $(x,y) \in f_n$ en een $m \in \mathbb{N}$ zo, dat $(x,z) \in f_m$. We mogen aannemen dat $n \leqslant m$.

Dan geldt $f_n \leqslant f_m$ en dus $(x,y) \in f_m$ en $(x,z) \in f_m$. Omdat $f_m \in T$, geldt dan $y = z$. Dus $f \in T$. De volgende bewering is dat $Pf = f$. Om dit te bewijzen stellen we $n \in \mathbb{N}$, dan $f_n \leqslant f$, dus $f_{n+1} = Pf_n \leqslant Pf$, maar ook $f_0 \leqslant Pf$ en dus $f \leqslant Pf$. Stel nu $x \in D_{Pf}$ en stel dat h de op grond van 2'. daarbij behorende partiële functie is, dus $h \in T_0$, $h \leqslant f$ en $x \in D_{Ph}$. Stel nu $y \in D_h$, dan $y \in D_f$, dus er is een $n \in \mathbb{N}$ zo, dat $y \in D_{f_n}$ en $f_n(y) = h(y)$; voor alle $m \geqslant n$ geldt dan ook $f_m(y) = h(y)$. Omdat D_h eindig is, zijn er maar eindig veel van zulke y; door van de bijbehorende n de grootste te nemen vinden we dat er een $n \in \mathbb{N}$ bestaat zo, dat $h \leqslant f_n$. Op grond van 2'. geldt dan $x \in D_{Pf_n} = D_{f_{n+1}} \subset D_f$. Daarmee is bewezen dat $D_{Pf} \subset D_f$. Tezamen met $f \leqslant Pf$ volgt daaruit $Pf = f$.

Stel nu een $g \in T$ waarvoor geldt $Pg = g$. Dan geldt ook $\forall_{n\in\mathbb{N}}\,[f_n \leqslant g]$. Dit bewijzen we met volledige inductie. $f_0 \leqslant g$ is triviaal. Stel $f_n \leqslant g$, dan $f_{n+1} = Pf_n \leqslant Pg = g$. Daaruit volgt dan weer $f \leqslant g$.

We noemen een $g \in T$, waarvoor geldt $Pg = g$ een <u>dekpunt</u> (Engels: fixed point) van P. De hierboven gevonden f is een dekpunt van P, maar bovendien geldt voor ieder dekpunt g van P, dat $f \leqslant g$. Daarom heet f <u>kleinste dekpunt</u> (Engels: least fixed point) van P. Dit geeft de volgende stelling.

<u>Stelling 12.1</u>. Stel A en B verzamelingen, T de verzameling der partiële functies van A naar B en $P: T \to T$, die aan voorwaarden 1. en 2. voldoet. Dan heeft P een kleinste dekpunt en wel $\cup\,\{P^n\emptyset \mid n \in \mathbb{N}\}$.

De bij een recursieve procedure verkregen functie is de partiële functie, die het kleinste dekpunt is van de bij de procedure behorende operator.

§13. PARTIEEL GEORDENDE VERZAMELINGEN

We willen overgaan tot een wat algemenere formulering, waarbij we niet meer aan partiële functies gebonden zijn. Het idee van vergroting van informatie willen we echter behouden, hetgeen uitgedrukt wordt in een partiële ordening. We beschouwen een partieel geordende verzameling $\langle S, \sqsubseteq \rangle$, dat is een verzameling S en een relatie $\sqsubseteq$ op S zo, dat voor alle $f \in S$, $g \in S$, $h \in S$ geldt

$$f \sqsubseteq f,$$

als $f \sqsubseteq g$ en $g \sqsubseteq h$, dan $f \sqsubseteq h$,

als $f \sqsubseteq g$ en $g \sqsubseteq f$, dan $f = g$.

Het blijkt, dat we om bruikbare resultaten te verkrijgen nog verdere restricties aan de partiële geordende verzameling zullen moeten opleggen. Welke restricties dat zijn, zullen we aan de analogie met partiële functies ontlenen. Eerst bespreken we echter enkele begrippen, die betrekking hebben op partieel geordende verzamelingen.

In een partieel geordende verzameling heet d een <u>grootste ondergrens</u> van f en g als $d \sqsubseteq f$, $d \sqsubseteq g$ en als uit $u \sqsubseteq f$ en $u \sqsubseteq g$ volgt $u \sqsubseteq d$.
Het is duidelijk, dat een aan deze voorwaarden voldoende d door f en g vastgelegd is zodat we van de grootste ondergrens van f en g mogen spreken, zo deze bestaat. We noteren $d = f \sqcap g$. In de verzameling der partiële functies hebben twee elementen f en g inderdaad een grootste ondergrens, namelijk de verzamelingstheoretische doorsnede $f \cap g$, die inderdaad een partiële functie is, die aan de vereisten voldoet.

Op analoge wijze definiëren we de kleinste bovengrens v van f en g door: f $\sqsubseteq$ v, g $\sqsubseteq$ v en uit f $\sqsubseteq$ u en g $\sqsubseteq$ u volgt v $\sqsubseteq$ u. We noteren v = f $\sqcup$ g. Bij partiële functies behoeft een kleinste bovengrens niet te bestaan. Stel dat er een a $\in D_f \cap D_g$ is, waarvoor geldt f(a) $\neq$ g(a). Voor een kleinste bovengrens zou moeten gelden f(a) = (f $\sqcup$ g)(a) en g(a) = (f $\sqcup$ g)(a), hetgeen onmogelijk is. Er bestaat dan zelfs geen gemeenschappelijke bovengrens h, laat staan een kleinste.

De begrippen $\sqcap$ en $\sqcup$ zijn in plaats van voor twee elementen ook voor een willekeurige deelverzameling D van S te formuleren. We doen het eerst voor $\sqcap$.

Stel D $\subset$ S, dan is $\sqcap$D het element van S dat voldoet aan $\forall_{f \in D}[\sqcap D \sqsubseteq f]$ en $\forall_{u \in S}[\forall_{f \in D}[u \sqsubseteq f] \Rightarrow u \sqsubseteq \sqcap D]$.
Voor een D $\subset$ S hoeft $\sqcap$D niet te bestaan, maar als hij bestaat is hij ondubbelzinnig vastgelegd. Nemen we D = $\emptyset$, dan voldoet $\sqcap\emptyset$ aan $\forall_{u \in S}[u \sqsubseteq \sqcap\emptyset]$. Dit is het grootste element van S, dat we, als het bestaat, met $\top$ (Engels: top) aanduiden.

De verzameling der partiële functies heeft geen grootste element, behalve in triviale uitzonderingsgevallen; om die uit te sluiten veronderstellen we voortaan dat #A $\geqslant$ 1 en #B $\geqslant$ 2. Voor iedere niet-lege D bestaat in dit geval echter wel $\sqcap$D, namelijk de verzamelingstheoretische doorsnede: $\sqcap D = \cap D$.

De definitie van $\sqcup$D gaat analoog. Ook is nu $\sqcup\emptyset$ het kleinste element van S, dat we, als het bestaat, met $\perp$ (Engels: bottom) aanduiden.

Voor partiële functies hoeft $\sqcup$D niet te bestaan, maar hij bestaat wel als

D een bovengrens heeft, dus als D een bovengrens heeft dan heeft D een kleinste bovengrens. Dit volgt gemakkelijk uit de volgende voor willekeurige partieel geordende verzamelingen geldende betrekking:

$$\bigsqcup D = \bigsqcap\{u \in S | \forall_{f \in D} [f \sqsubseteq u]\} .$$

Dit moet zo gelezen worden, dat als $\bigsqcap$ in het rechterlid bestaat, dan $\bigsqcup$ in het linkerlid ook bestaat en de leden aan elkaar gelijk zijn. Deze betrekking is gemakkelijk aan te tonen. We kunnen dit toepassen op de verzameling T van partiële functies en een deelverzameling D van T die een bovengrens heeft. Dan is de verzameling in het rechterlid niet leeg, maar daaruit volgt in $\bar{T}$ dat $\bigsqcap$ van die verzameling bestaat.

De verzameling T voldoet echter aan een sterkere voorwaarde voor bovengrenzen, namelijk: als $D \subset T$ en ieder tweetal elementen van D heeft een bovengrens in T, dan heeft D een bovengrens in T.

<u>Stelling 13.1</u>. Stel T de verzameling van alle partiële functies van A naar B, $D \subset T$, $\forall_{f \in D} \forall_{g \in D} \exists_{v \in T} [f \leqslant v$ en $g \leqslant v]$. Dan bestaat er een $u \in T$ zo, dat $\forall_{f \in D} [f \leqslant u]$.

<u>Bewijs</u>: Kies $u = \cup D$. Het is voldoende om te bewijzen dat u een partiële functie is. Stel $(x,y) \in u$ en $(x,z) \in u$ dan is er een $f \in D$ waarvoor $y = f(x)$ en een $g \in D$ waarvoor $z = g(x)$. Volgens veronderstelling is er nu ook een $v \in T$ zo, dat $f \leqslant v$ en $g \leqslant v$, maar dan is $y = f(x) = v(x) = g(x) = z$. ∎

Een iets zwaardere eis voor D krijgen we, als we $v \in D$ eisen. Dit leidt tot het begrip gerichte verzameling (Engels: directed set).

Definitie gerichte verzameling

Een partieel geordende verzameling $\langle S, \sqsubseteq \rangle$ heet gericht als

$\forall_{f \in S} \forall_{g \in S} \exists_{v \in S} [f \sqsubseteq v \text{ en } g \sqsubseteq v]$.

We passen dit vaak toe op een deelverzameling D van een gegeven partieel geordende verzameling S; in dat geval wordt de partiële ordening van D ontleend aan die van S; de v in de definitie moet dan ook in D liggen. De lege verzameling voldoet aan de eisen en is dus een gerichte verzameling; in de literatuur wordt hier vaak een uitzondering gemaakt en de lege verzameling niet meegeteld als gerichte verzameling; wij zullen haar echter wel meetellen.

Voorbeelden van gerichte deelverzamelingen van S zijn totaal geordende deelverzamelingen (een partieel geordende verzameling S heet totaal geordend als $\forall_{f \in S} \forall_{g \in S} [f \sqsubseteq g \text{ of } g \sqsubseteq f]$). Een speciaal geval van een totaal geordende verzameling is een niet-dalende rij: $\{f_n | n \in \mathbb{N}\}$ waarvoor geldt $\forall_{n \in \mathbb{N}} [f_n \sqsubseteq f_{n+1}]$. Een verzameling die een grootste element heeft, is uiteraard ook gericht.

De verzameling T van partiële functies voldoet nu aan de volgende twee voorwaarden (geformuleerd voor een partieel geordende verzameling S).

I. $\sqcup D$ bestaat voor alle gerichte $D \subset S$.

II. $\sqcap D$ bestaat voor alle niet-lege $D \subset S$.

Als in een partieel geordende verzameling $\sqcap D$ bestaat voor alle $D \subset S$, inclusief $D = \emptyset$, dan heet S een volledig tralie (Engels: complete lattice). In dat geval bestaat ook $\sqcup D$ voor alle $D \subset S$ en S heeft een grootste element $\top = \sqcap \emptyset$ en een kleinste element $\bot = \sqcup \emptyset$. Er geldt ook $\sqcup S = \top$ en $\sqcap S = \bot$.

De verzameling der partiële functies is geen volledig tralie, maar kan tot een volledig tralie uitgebreid worden door een element $\top$ toe te voegen en te verklaren, dat $f \leqslant \top$ voor alle $f \in T$. Voor de $D \subset T$, waarvoor $\bigsqcup D$ in T niet bestaat, geldt dan $\bigsqcup D = \top$ in de grotere verzameling. We zullen van deze uitbreiding geen gebruik maken.

We gaan nu verder met een partieel geordende verzameling die aan voorwaarde I. voldoet; het blijkt niet nodig te zijn om voorwaarde II. te eisen en we zullen dat daarom ook niet doen. Een dergelijke partiële verzameling heet een cpo (Engels: complete partial order).

Definitie cpo

Een partieel geordende verzameling $\langle S, \sqsubseteq \rangle$ met de eigenschap, dat $\bigsqcup D$ bestaat voor alle gerichte $D \subset S$, heet een cpo.

Omdat $\bigsqcup \emptyset$ ook bestaat in een cpo, heeft een cpo een kleinste element $\perp$. Een verzameling T van partiële functies is een cpo; een volledig tralie is ook een cpo.

We beschouwen nu een operator $P: S \to S$, waarbij S een cpo is en vragen ons af welke beperkingen we aan P zullen opleggen. Van de voorwaarden 1. en 2. die we bij partiële functies hebben opgelegd is 1. (monotonie) direct naar S over te dragen. Voor 2. is dat echter niet duidelijk. We leggen nu een geheel anders ogende voorwaarde aan P op en zullen later (Stelling 13.4) bewijzen dat deze in het geval van partiële functies equivalent is met voorwaarden 1. en 2. Een operator die aan deze voorwaarden voldoet heet continu.

Definitie continue operator

Stel $\langle S,\sqsubseteq\rangle$ een cpo en $P\colon S \to S$. Dan heet P continu als geldt: voor alle $D \subset S$, D niet-leeg en gericht, bestaat $\sqcup\{Pf \mid f \in D\}$ en $\sqcup\{Pf \mid f \in D\} = P(\sqcup D)$.

Stelling 13.2. Een continue operator P op een cpo heeft een kleinste dekpunt en wel $\sqcup\{P^n\bot \mid n \in \mathbb{N}\}$.

Bewijs: Stel $\langle S,\sqsubseteq\rangle$ een cpo en P continu. We bewijzen eerst, dat P monotoon is. Stel $f \in S, g \in S$, $f \sqsubseteq g$. Dan is $\{f,g\}$ een niet-lege gerichte verzameling. Omdat P continu is bestaat $\sqcup\{Ph \mid h \in \{f,g\}\} = Pf \sqcup Pg$ en dit is $= P(\sqcup\{f,g\}) = Pg$. Daaruit volgt $Pf \sqsubseteq Pg$. Definieer nu de rij $f_0, f_1, \ldots$ door $f_0 = \bot$ en $f_{n+1} = Pf_n$. Dan geldt $\forall_{n\in\mathbb{N}}[f_n \sqsubseteq f_{n+1}]$, omdat P monotoon is. Dus $\{f_n \mid n \in \mathbb{N}\}$ is een niet-lege, gerichte verzameling. Omdat S een cpo is, bestaat $p = \sqcup\{f_n \mid n \in \mathbb{N}\}$ en omdat P continu is, geldt $Pp = \sqcup\{Pf_n \mid n \in \mathbb{N}\} = \sqcup\{f_{n+1} \mid n \in \mathbb{N}\} = p$, dus p is dekpunt van P. Stel nu $q \in S$, $Pq = q$, dan geldt $\forall_{n\in\mathbb{N}}[f_n \sqsubseteq q]$. Dit bewijzen we met volledige inductie: $f_0 = \bot \sqsubseteq q$; als $f_n \sqsubseteq q$, dan $f_{n+1} = Pf_n \sqsubseteq Pq = q$. Dus is g een bovengrens van $\{f_n \mid n \in \mathbb{N}\}$, dus $p \sqsubseteq q$. ∎

We bespreken nu enkele varianten op Stelling 13.2. Een klassieke stelling (Knaster-Tarski) zegt, dat in een volledig tralie een monotone operator een kleinste dekpunt heeft. Voor de door ons beoogde toepassingen heeft dit het bezwaar, dat we vaak niet met volledige tralies te maken hebben. Bij partiële functies hebben we bij uitbreiding tot een volledig tralie geen garantie dat het dekpunt niet het toegevoegde element $\top$ is. Er bestaan ook uitbreidingen van Stelling 13.2, die ontstaan door bij gelijkblijvende conclusie de veronderstellingen te verzwakken. Men kan de eis dat S een cpo is vervangen door de eis, dat $\sqcup D$ bestaat voor alle totaal geordende $D \subset S$. Als men dit doet, wordt het dubieus hoe een continue operator moet worden gedefinieerd. We maken ons daarover echter niet druk, omdat de genoemde verzwakking geen echte verzwakking blijkt te zijn. Er geldt namelijk de volgende stelling.

Stelling 13.3. Als in een partieel geordende verzameling $\langle S,\sqsubseteq\rangle$ geldt, dat $\sqcup D$ bestaat voor alle totaal geordende $D \subset S$, dan bestaat $\sqcup D$ ook voor alle gerichte $D \subset S$.

Het (moeilijke) bewijs van deze stelling behandelen we niet. We kunnen ook de eis voor P drastisch verzwakken. Het blijkt zo te zijn, dat in een cpo iedere monotone operator P een kleinste dekpunt heeft. Ook dit bewijzen we niet; het bewijs is moeilijker dan dat van Stelling 13.2, maar bovendien ook minder constructief, wat onder meer inhoudt, dat er niet zo'n mooie uitdrukking voor het kleinste dekpunt uit komt, hetgeen de praktische toepasbaarheid van het resultaat nadelig beïnvloedt.

We tonen nu nog aan, dat voor partiële functies continuïteit van een operator overeenkomt met de voorwaarden 1. en 2.

Stelling 13.4. Stel A en B verzamelingen, T de verzameling van alle partiële functies van A naar B en $P: T \to T$. Dan geldt:

$$P \text{ voldoet aan 1. en 2.} \Leftrightarrow P \text{ is continu.}$$

Bewijs: We hebben eerder bewezen, dat T een cpo is en dat als $\sqcup D$ bestaat voor $D \subset T$, dan $\sqcup D = \cup D$. We veronderstellen nu, dat P voldoet aan 1. en 2. Stel $D \subset T$, D niet-leeg en gericht. Uit 1. volgt, dat ook $\{Pf \mid f \in D\}$ gericht is en dus dat $\sqcup\{Pf \mid f \in D\}$ bestaat. Maar $\sqcup D$ bestaat ook en $\sqcup D = \cup D$. Stel nu $f \in D$, dan $f \leqslant \sqcup D$, dus $Pf \leqslant P(\sqcup D)$. Dus $\sqcup\{Pf \mid f \in D\} \leqslant P(\sqcup D)$. Stel nu $(x,y) \in P(\sqcup D)$. Op grond van 2'. is er een $h \in T_0$ zo, dat $h \leqslant \sqcup D$ en $(x,y) \in Ph$. Stel nu $(z,w) \in h$, dan is er, wegens $h \subset \cup D$, een $k \in D$ waarvoor $(z,w) \in k$. Omdat h eindig is en D niet-leeg en gericht volgt hieruit, dat er een $k \in D$ bestaat zo, dat $h \subset k$ (let wel dat als $h = \emptyset$, de voorwaarde $D \neq \emptyset$ gebruikt wordt). Maar op grond van 1. geldt ook $Ph \subset Pk$ en $(x,y) \in Pk \subset \sqcup\{Pf \mid f \in D\}$. Daarmee is aangetoond, dat $P(\sqcup D) \leqslant \sqcup\{Pf \mid f \in D\}$. Beide ongelijkheden samen geven dat $P(\sqcup D) = \sqcup\{Pf \mid f \in D\}$. Dus P is continu.
Nu gaan we omgekeerd van de veronderstelling uit, dat P continu is. Het bewijs dat 1. geldt, ontlenen we aan het bewijs van Stelling 13.2. Ten behoeve van 2. merken we eerst op dat als $f \in T$, dan $\{h \in T_0 \mid h \leqslant f\}$ niet-leeg en gericht, en $\sqcup\{h \in T_0 \mid h \leqslant f\} = \cup\{h \in T_0 \mid h \leqslant f\} = f$. Uit de continuïteit van P volgt dan, dat $Pf = \sqcup\{Ph \mid h \in T_0 \text{ en } h \leqslant f\} = \cup\{Ph \mid h \in T_0 \text{ en } h \leqslant f\}$. Stel nu $x \in D_{Pf}$, dan is er een $h \in T_0$ zo, dat $h \leqslant f$ en $(x,(Pf)(x)) \in Ph$ en dus $x \in D_{Ph}$. Daarmee is 2". en dus 2. aangetoond. ∎

We merken nog op dat uit bovenstaand bewijs volgt dat voor een continue operator P op een verzameling T van partiële functies en een $f \in T$ geldt:

$$Pf = \sqcup\{Ph \mid h \in T_0 \text{ en } h \leqslant f\} ,$$

hetgeen op zichzelf ook een interessante betrekking is. Er volgt uit, dat door de Ph voor $h \in T_0$ alle Pf voor $f \in T$ vastgelegd zijn. Dit brengt ons op de vraag of een operator P_0 die slechts op T_0 gedefinieerd is voortgezet kan worden tot een continue operator P op T. Uit het bovenstaande volgt dat als het kan, het op slechts één manier kan. Het is duidelijk, dat het noodzakelijk is, dat P_0 monotoon is. Dit blijkt echter ook voldoende te zijn.

Stelling 13.5. Als T de verzameling der partiële functies $A \to B$ is en $P_0 \colon T_0 \to T$ een monotone operator, dan is $P \colon T \to T$ gedefinieerd door

$$Pf := \bigcup\{P_0 h \mid h \in T_0 \text{ en } h \leqslant f\} \text{ voor } f \in T,$$

een continue operator $T \to T$ en $\forall_{f \in T_0}[Pf = P_0 f]$.

Bewijs: In de eerste plaats is Pf, zoals in de stelling gedefinieerd, een partiële functie. Dit is zo, omdat uit de monotonie van P_0 volgt, dat $\{P_0 h \mid h \in T_0 \text{ en } h \leqslant f\}$ een gerichte verzameling is. Stel nu $f \in T_0$, $h \in T_0$, $h \leqslant f$; dan $P_0 h \leqslant P_0 f$ en dus $Pf = \bigcup\{P_0 h \mid h \in T_0 \text{ en } h \leqslant f\} = P_0 f$. Daarmee is $\forall_{f \in T_0}[Pf = P_0 f]$ aangetoond.

We bewijzen nu 1. en 2. voor P.

Stel $f \in T$, $g \in T$, $f \leqslant g$, dan $\{P_0 h \mid h \in T_0 \text{ en } h \leqslant f\} \subset \{P_0 h \mid h \in T_0 \text{ en } h \leqslant g\}$, dus $\bigcup\{P_0 h \mid h \in T_0 \text{ en } h \leqslant f\} \leqslant \bigcup\{P_0 h \mid h \in T_0 \text{ en } h \leqslant g\}$, dus $Pf \leqslant Pg$. Daarmee is 1. aangetoond. Stel nu $(x,y) \in Pf$, dan is er een $h \in T_0$ zo, dat $h \leqslant f$ en $(x,y) \in P_0 h = Ph$. Dus als $x \in D_{Pf}$, dan is er een $h \in T_0$ zo, dat $h \leqslant f$ en $x \in D_{Ph}$. Daarmee is 2". en dus 2. aangetoond. Dus P is continu. ∎