Recursietheorie

Jaap van Oosten
Opleidingsinstituut Wiskunde
Universiteit Utrecht

1993, herzien 2001

Contents

Inleiding

De recursietheorie houdt zich bezig met de studie van functies en relaties op de verzameling $\mathbb{N}$ der natuurlijke getallen (d.w.z. $f : \mathbb{N}^k \to \mathbb{N}$ en $A \subseteq \mathbb{N}^k$) vanuit het standpunt van *berekenbaarheid*: gegeven $f : \mathbb{N}^k \to \mathbb{N}$, of $A \subseteq \mathbb{N}^k$, is er een algoritme waarmee ik voor elke $\vec{n} = (n_1, \ldots, n_k) \in \mathbb{N}^k$ $f(\vec{n})$ kan berekenen, resp. bepalen of $\vec{n} \in A$?

Dit soort vragen zijn uiteraard pas voor wiskundige behandeling vatbaar als we aan het begrip "algoritme" een precieze betekenis hebben gegeven. Voorlopig kun je denken aan: computerprogramma. Maar het woord 'computerprogramma' veronderstelt een bepaalde programmeertaal, zoals PASCAL, C++, Haskell... Wij zullen ons niet bezighouden met wat er in specifieke programmeertalen te programmeren valt: het is de pretentie van de Recursietheorie, te bestuderen wat er in *welke programmeertaal dan ook* te programmeren valt.

Een ander aspect van "programmeren" dat we buiten beschouwing zullen laten, is de vraag: hoe efficiënt werkt een gegeven algoritme? Dit is de vraag die centraal staat in het vak *Complexiteitstheorie*, dat bij Informatica wordt gegeven.

Nadat we hebben gedefinieerd wat we onder een (algoritmisch) berekenbare functie verstaan, zullen we zien dat al deze functies in een definitieschema zijn onder te brengen, waarin het begrip 'recursie' een grote rol speelt.

Het begrip 'berekenbaar' kan ook gerelativeerd worden. De vraag kan dan luiden voor gegeven functies f en g: kan ik een algoritme voor f geven in termen van de functiewaarden van g? Het is dan, alsof we f kunnen berekenen mits een "orakel" ons vertelt wat g doet, en we noemen f "berekenbaar in g". Dit gezichtspunt komt in dit college ook aan de orde (hoofdstuk 4).

De meest direkte toepassingen van de recursietheorie zijn van theoretische aard, zoals:

- Er is geen algoritme dat voor een willekeurig polynoom $P(X_1, \ldots, X_n)$ met coëfficiënten in $\mathbb{Z}$ vaststelt of P een nulpunt heeft in $\mathbb{Z}^n$;
- Er is geen algoritme dat voor willekeurige predicaatlogische formules φ vaststelt of φ te bewijzen is in de predicatenlogica.

In dit college zal tevens de beroemde *Eerste Onvolledigheidsstelling van Gödel* aan de orde komen, die grofweg zegt dat geen enkel redelijk overzichtelijk axiomasysteem in een taal die minstens optelling en vermenigvuldiging bevat, *alle* ware uitspraken over $\mathbb{N}$ kan bewijzen.

Systematisch nadenken over algoritmen stamt vooral uit de jaren '30, als reactie op Gödels onvolledigheidsbewijs uit 1931. De belangrijkste namen zijn Alan Turing, Alonzo Church en Stephen Cole Kleene. Alle drie bedachten ze een formalisering van het intuïtieve begrip "algoritme" en alle drie de formaliseringen zijn equivalent in de zin dat ze dezelfde klasse van algoritmisch berekenbare functies definiëren. Wij bestuderen in deze syllabus een andere, ook equivalente, benadering, die dichter bij het intuïtief rekenen ligt: het concept "registermachine".

Aanbevolen literatuur over recursietheorie: hèt standaardwerk is nog altijd

H. Rogers, *Theory of recursive functions and effective computability*, McGraw-Hill 1967

hoewel vrij oud en niet altijd even prettig leesbaar.

Een moderner leerboek is:

P. Odifreddi, *Classical Recursion Theory*, North-Holland (Amsterdam) 1989,

helaas nogal duur.

Een erg aardig boek waarin het grootste deel van de stof van dit college wordt behandeld (hoewel de nadruk op andere dingen ligt) en dat veel achtergrond geeft, met name voor hen die in logica geïnteresseerd zijn, is:

C. Smoryński, *Logical Number Theory I, An Introduction*, Springer 1991 (Universitext)

Ook aan te bevelen (vooral voor onderwerpen in het verlengde van hoofdstukken 3 en 4 van dit diktaat):

R.I. Soare, *Recursively enumerable sets and degrees*, Springer 1987

Een beknopt overzicht geeft

J.R. Shoenfield, *Recursion Theory*, Springer 1993 (Lecture Notes in Logic 1)

Afspraak. In dit dictaat beginnen de natuurlijke getallen bij 0: $\mathbb{N} = \{0, 1, 2, \ldots\}$

1 Berekenbare Functies

1.1 Registermachines

Wij denken aan een geïdealiseerde "computer", een *registermachine*, met een oneindig aantal geheugenplaatsen, *registers* geheten, die elk een willekeurig groot natuurlijk getal kunnen bevatten. De registers zijn genummerd $R_1, \ldots$

Programma's voor de registermachine worden geschreven in een minuscule programmeertaal met slechts twee soorten instructies (een programma is een *genummerde* lijst instructies):

$r_i^+ \Rightarrow j$	wordt gelezen als:	tel één op bij de inhoud van R_i en ga naar instructie j
$r_i^- \Rightarrow j, k$	wordt gelezen als:	als de inhoud van R_i groter is dan 0, trek er één van af en ga naar instructie j; anders, ga naar instructie k

Opmerkingen: het kan voorkomen dat de machine naar instructie k gestuurd wordt terwijl er geen instructie k in het programma voorkomt; de machine stopt dan. We nemen aan dat de instructies genummerd van 1 t/m k onder elkaar staan; $k + 1$ is dan de eerste niet-bestaande instructie en kan dienst doen als stop-instructie.

Uiteraard zijn programma's altijd *eindig*.

Voorbeelden

a)	$1 : r_i^+ \Rightarrow 2$	telt één op bij de inhoud van R_i en stopt dan
b)	$1 : r_i^+ \Rightarrow 1$	blijft voortdurend één bij R_i optellen
c)	$1 : r_i^- \Rightarrow 1, 2$	register R_i wordt "geleegd"
d)	$1 : r_i^- \Rightarrow 2, 3$	de inhoud van R_i wordt overgebracht naar R_j
	$2 : r_j^+ \Rightarrow 1$	
e)	$1 : r_i^- \Rightarrow 2, 4$	de inhoud van R_i wordt tegelijkertijd naar R_j en R_k
	$2 : r_j^+ \Rightarrow 3$	overgeheveld
	$3 : r_k^+ \Rightarrow 1$	

We nemen nu voorts aan, dat in de registermachine altijd slechts eindig veel registers een inhoud groter dan 0 hebben. Omdat elk programma eindig is en dus maar eindig veel registers noemt (aanroept), zijn er op ieder moment in een berekening slechts eindig veel registers bezet.

We kunnen nu een formele definitie geven van een *berekening* van de registermachine met programma P:

Definitie 1.1 Als P een programma is, is een *berekening* van de registermachine met programma P een eindige of oneindige rij geordende $k + 1$-tallen ($k \geq 1$) $(n_i, r_1^i, \ldots, r_k^i)$ voor $i \geq 1$, zodat:

1. $n_1 = 1$
2. als $n_i = m$, dan één van de volgende mogelijkheden:
   - P heeft geen instructie m en $(n_i, r_1^i, \ldots, r_k^i)$ is de laatste term van de rij (de rij is eindig);
   - De m-de instructie van P is $r_j^+ \Rightarrow l,\ j \leq k,\ n_{i+1} = l$ en $(r_1^{i+1}, \ldots, r_k^{i+1}) = (r_1^i, \ldots, r_j^i + 1, \ldots, r_k^i)$

- De m-de instructie van P is $r_j^- \Rightarrow l, p$, $j \leq k$ en nu:
  $r_j^i > 0$ en $(n_{i+1}, r_1^{i+1}, \ldots, r_k^{i+1}) = (l, r_1^i, \ldots, r_j^i - 1, \ldots, r_k^i)$, of
  $r_j^i = 0$ en $(n_{i+1}, r_1^{i+1}, \ldots, r_k^{i+1}) = (p, r_1^i, \ldots, r_k^i)$

Definitie 1.2 Zij $F : A \to \mathbb{N}$ waar $A \subseteq \mathbb{N}^k$. We noemen F een *partiële functie* en schrijven ook wel $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$. We noemen A het *domein* van F en schrijven $A = \mathrm{dom}(F)$.

F heet *registermachine berekenbaar* of kortweg *berekenbaar* als er een registermachine programma P is en een $s \geq 1$ zodat voor alle $(m_1, \ldots, m_k) \in \mathbb{N}^k$ geldt: als $(m_1, \ldots, m_k) \in A$ dan is er een *eindige* berekening

$$(n_i, r_1^i, \ldots, r_{k+s}^i)_{i=1}^K$$

van de registermachine met programma P, zodat geldt:

- $(r_1^1, \ldots, r_{k+s}^1) = (0, m_1, \ldots, m_k, 0, \ldots, 0)$
- $(r_1^K, \ldots, r_{k+s}^K) = (F(m_1, \ldots, m_k), m_1, \ldots, m_k, 0, \ldots, 0)$

Notatie. We schrijven berekeningen vaak zó op:

$$\begin{pmatrix} a_1\ a_2 \cdots a_m \\ \Downarrow\ P \\ b_1\ b_2 \cdots b_n \end{pmatrix}$$

waar $a_1, \ldots, a_m$ de registerinhouden van de niet-lege registers zijn vòòr de berekening met programma P, en $b_1, \ldots, b_n$ die erna. Een paar vaak gebruikte programma's geven we een standaardaanduiding:
dat van voorbeeld c) van blz. 3 (R_i legen) noemen we L_i;
dat van voorbeeld d) van blz. 3 (R_i in R_j kopiëren) noemen we $K_{i,j}$;
dat van voorbeeld e) van blz. 3 (R_i in R_j en R_k kopiëren) noemen we $K_{i,j,k}$.

Samenstellen van programma's
Als

$$\begin{pmatrix} a_1\ a_2 \cdots a_m \\ \Downarrow\ P \\ b_1\ b_2 \cdots b_n \end{pmatrix} \text{ en } \begin{pmatrix} b_1\ b_2 \cdots b_n \\ \Downarrow\ Q \\ c_1\ c_2 \cdots c_r \end{pmatrix}$$

berekeningen zijn, hebben we ook een berekening:

$$\begin{pmatrix} a_1\ a_2 \cdots a_m \\ \Downarrow\ PQ \\ c_1\ c_2 \cdots c_r \end{pmatrix}$$

waar PQ het programma is dat ontstaat door Q achter P te zetten, de nummers van alle instructies in Q met K te verhogen (waar K het hoogste instructienummer is dat in P voorkomt; dus bijv. $i : r_j^+ \Rightarrow k$ wordt $i+K : r_j^+ \Rightarrow k+K$), en indien nodig overal waar in P naar een instructienummer $> K$ verwezen wordt, van dit nummer $K+1$ te maken. In het vervolg zullen we dus samenstellingen als

$$\begin{pmatrix} \vec{a} \\ \Downarrow\ P \\ \vec{b} \\ \Downarrow\ Q \\ \vec{c} \\ \Downarrow\ R \\ \vec{d} \end{pmatrix}$$

als één berekening beschouwen.

Propositie 1.3 *Stel* $G_1, \ldots, G_l : \mathbb{N}^k \rightharpoonup \mathbb{N}$ *en* $F : \mathbb{N}^l \rightharpoonup \mathbb{N}$ *zijn berekenbare functies. Dan is ook de* compositie H*, gedefinieerd door*

$$H(\vec{x}) = F(G_1(\vec{x}), \ldots, G_l(\vec{x}))$$

berekenbaar.

Bewijs. H is uiteraard een *partiële* functie: het domein van H is

$$\{\vec{x} \in \mathbb{N}^k \mid \vec{x} \in \bigcap_{i=1}^{l} \mathrm{dom}(G_i) \ \wedge \ (G_1(\vec{x}), \ldots, G_l(\vec{x})) \in \mathrm{dom}(F)\}$$

Het bewijs wordt gegeven door het volgende diagram, waarbij we aannemen dat de programma's $P_1, \ldots, P_l$ en Q resp. de functies $G_1, \ldots, G_l$ en F berekenen op de manier als in definitie 1.2, en dat N een getal is groter dan k en l en groter dan elk registernummer dat voorkomt in $P_1, \ldots, P_l$ of Q:

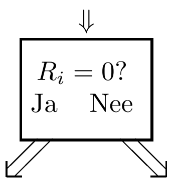
$$\begin{array}{l}
0\ \vec{x}\ \vec{0} \\
\Downarrow\ P_1 \\
G_1(\vec{x})\ \vec{x}\ \vec{0} \\
\Downarrow\ K_{1,N+1} \\
0\ \vec{x}\ \vec{0}\ G_1(\vec{x}) \\
\Downarrow\ P_2 \\
G_2(\vec{x})\ \vec{x}\ \vec{0}\ G_1(\vec{x}) \\
\Downarrow \\
\vdots \\
\Downarrow \\
G_l(\vec{x})\ \vec{x}\ \vec{0}\ G_1(\vec{x}) \cdots G_{l-1}(\vec{x}) \\
\Downarrow\ K_{1,N+l} \\
0\ \vec{x}\ \vec{0}\ G_1(\vec{x}) \cdots G_l(\vec{x}) \\
\Downarrow\ K_{2,N+l+1} \cdots K_{k+1,N+l+k}\ K_{N+1,2} \cdots K_{N+l,l+1} \\
0\ G_1(\vec{x}) \cdots G_l(\vec{x})\ \vec{0}\ \vec{x} \\
\Downarrow\ Q \\
H(G_1(\vec{x}), \ldots, G_l(\vec{x}))\ G_1(\vec{x}) \cdots G_l(\vec{x})\ \vec{0}\vec{x} \\
\Downarrow\ L_2 \cdots L_{l+1}\ K_{N+l+1,2} \cdots K_{N+l+k,k+1} \\
H(G_1(\vec{x}), \ldots, G_l(\vec{x}))\ \vec{x}\ \vec{0}
\end{array}$$

■

We kunnen in registermachineprogramma's blokjes aantreffen van de vorm:

$$\begin{array}{rcl}
k : r_i^- & \Rightarrow & k+1, l \\
k+1 : r_i^+ & \Rightarrow & m
\end{array}$$

waarvan de werking die is van *gevalsonderscheiding*: als R_i leeg is, ga naar instructie l, en anders naar instructie m. In onze diagrammen kunnen we dit weergeven door



waarbij we verschillende programma's bij de twee pijlen Ja en Nee kunnen plaatsen.

Propositie 1.4 *Stel $G : \mathbb{N}^k \rightharpoonup \mathbb{N}$ berekenbaar en $H : \mathbb{N}^{k+2} \rightharpoonup \mathbb{N}$ berekenbaar. Dan is ook de functie $F : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ berekenbaar, die gedefinieerd is als volgt:*

$$F(0,\vec{x}) = G(\vec{x})$$

$$F(y+1,\vec{x}) = H(y, F(y,\vec{x}), \vec{x})$$

Opmerkingen: Ook F is een partiële functie; het domein van F is

$$\{(y,\vec{x}) \mid \vec{x} \in \mathrm{dom}(G) \wedge \forall w < y((w, F(w,\vec{x}), \vec{x}) \in \mathrm{dom}(H))\}$$

Ga zelf na dat dit een zinvolle definitie is.

F heet gedefinieerd uit G en H door middel van *primitieve recursie*.

Bewijs. Het zal je niet verbazen dat het diagram een 'loop' bevat:

$0\ y\ \vec{x}\ \vec{0}$
$\Downarrow$
$0\ \vec{x}\ \vec{0}\ y\ y\ \vec{x}$
$\Downarrow\ P$
$G(\vec{x})\ \vec{x}\ \vec{0}\ y\ y\ \vec{x}$
$\Downarrow$
$\vdots$
$F(c,\vec{x})\ \vec{x}\ c\ \vec{0}\ y-c\ y\ \vec{x}$
$\Downarrow$
$\boxed{R_i = 0?\ \text{Ja}\quad\text{Nee}}$

Ja: $F(c,\vec{x})\ c\ \vec{x}\ \vec{0}$
STOP

Nee: $0\ c\ F(c,\vec{x})\ \vec{x}\ \vec{0}\ y-c\ y\ \vec{x}$
$\Downarrow\ Q$
$F(c+1,\vec{x})\ c\ F(c,\vec{x})\ \vec{x}\ \vec{0}\ y-c\ y\ \vec{x}$
$\Downarrow$
$F(c+1,\vec{x})\ \vec{x}\ c\ \vec{0}\ y-c\ y\ \vec{x}$
$\Downarrow$
$F(c+1,\vec{x})\ \vec{x}\ c+1\ \vec{0}\ y-(c+1)\ y\ \vec{x}$

Hierbij zijn P en Q programma's die resp. G en H berekenen. De niet-benoemde pijlen staan voor evidente combinaties van kopiëren en leeg maken, optellen en aftrekken. In de toestand van de machine juist vòòr de gevalsonderscheiding staat $y-c$ in het i-de register

∎

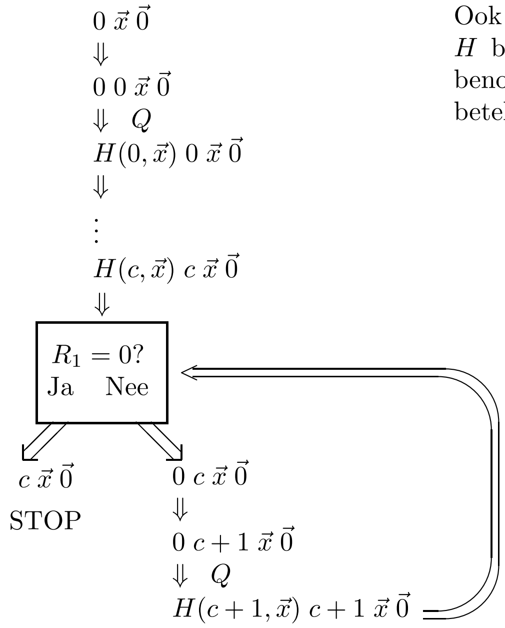
Propositie 1.5 *Laat $H : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ een partiële functie zijn met domein $A \subseteq \mathbb{N}^{k+1}$. Definieer de partiële functie $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ als volgt: het domein van F is*

$$\{\vec{x} \in \mathbb{N}^k \mid \exists y \in \mathbb{N}\, \forall w \leq y\, ((w,\vec{x}) \in A \wedge H(y,\vec{x}) = 0)\}$$

en voor $\vec{x}$ in het domein van F is $F(\vec{x})$ de minimale zulke y.
Dan geldt: als H berekenbaar is, is F het ook.

F heet gedefinieerd uit H door middel van *minimalisatie*.

Bewijs.



Ook hier is Q een programma dat H berekent en hebben de niet-benoemde pijlen hun evidente betekenis

∎

Proposities 1.3, 1.4 en 1.5 tonen aan dat zeer veel functies berekenbaar zijn.

Opgave 1. Schrijf registermachine-programma's die de volgende functies berekenen:

i) $F(x, y) = x \dot{-} y$ (het afgeknotte verschil van x en y: $x \dot{-} y$ is $x - y$ als $x \geq y$, en 0 anders)

ii) $F(x, y) = 0$ als $x|y$, 1 anders

iii) $F(x, y) = \mathrm{ggd}(x, y)$

iv) $F(x, y) =$ rest van y bij deling door x

v) $F(x, y) = xy$

vi) $F(x, y) = x^y$

vii) $F(x_1, \ldots, x_k) = x_i \ \ (1 \leq i \leq k)$

viii) $F(x_1, \ldots, x_k) = n$ (n vast)

Het is natuurlijk a priori niet gezegd dat alle partiële functies $\mathbb{N}^k \rightharpoonup \mathbb{N}$ waarvoor men een algoritme kan verzinnen, registermachine-berekenbaar zijn. De uitspraak dat dit wel zo is, staat bekend als de *These van Church*. Tot op heden is niemand erin geslaagd een algoritme te geven voor een niet-registermachine-berekenbare functie.

1.2 Primitief recursieve functies en relaties

Notatie van functies. In wiskundige teksten is het gebruikelijk om uitdrukkingen met variabelen als $x + y$, x^2, $x \log(y)$ etc. zowel te gebruiken als aanduiding van een (variabel) *getal*, als van de *functie* van de erin voorkomende variabelen. In de recursietheorie, die voor een groot deel gaat over manieren om functies te definiëren, is het van belang dit soort verschillende betekenissen ook qua notatie uit elkaar te houden. We kunnen de uitdrukking $x \log(y)$ bijvoorbeeld opvatten:

- als getal

- als functie van (x, y), dus als functie: $\mathbb{R}^2 \to \mathbb{R}$
- als functie van (y, x), dus als een andere functie: $\mathbb{R}^2 \to \mathbb{R}$
- als functie van y (met variabele x, dus eigenlijk een serie functies: $\mathbb{R} \to \mathbb{R}$)
- als functie van (x, y, z), dus: $\mathbb{R}^3 \to \mathbb{R}$

Om dit te onderscheiden gebruiken we de zg. λ-notatie: als $\vec{x}$ een rijtje variabelen $x_1 \cdots x_k$ is die al of niet kunnen voorkomen in een uitdrukking G, stelt $\lambda\vec{x}.G$ de functie voor die aan het k-tal $n_1 \cdots n_k$ de waarde $G(n_1, \ldots, n_k)$ toekent (de n_i voor de x_i in G substitueren). Met deze notatie schrijven we de vijf bovenstaande gevallen zó op: $x \log(y)$, $\lambda xy.x \log(y)$, $\lambda yx.x \log(y)$, $\lambda y.x \log(y)$ en $\lambda xyz.x \log(y)$.

Definitie 1.6 De klasse van *primitief recursieve* functies $\mathbb{N}^k \to \mathbb{N}$ (voor *variabele* k) wordt voortgebracht door de volgende clausules:

i) de *nulfunctie* $N = \lambda x.0$ is primitief recursief;

ii) de *opvolgerfunctie* $S = \lambda x.x + 1$ is primitief recursief;

iii) de *projectiefuncties* $\Pi^k_i = \lambda x_1 \cdots x_k.x_i$ (voor $1 \leq i \leq k$) zijn primitief recursief;

iv) Als $G_1, \ldots, G_l : \mathbb{N}^k \to \mathbb{N}$ en $H : \mathbb{N}^l \to \mathbb{N}$ primitief recursief zijn, dan is ook $\lambda\vec{x}.H(G_1(\vec{x}), \ldots, G_l(\vec{x}))$ primitief recursief;

v) Als $G : \mathbb{N}^k \to \mathbb{N}$ en $H : \mathbb{N}^{k+2} \to \mathbb{N}$ primitief recursief zijn, dan ook de functie F gedefinieerd uit G en H door middel van primitieve recursie:

$$\begin{array}{rcl} F(0, \vec{x}) & = & G(\vec{x}) \\ F(y + 1, \vec{x}) & = & H(y, F(y, \vec{x}), \vec{x}) \end{array}$$

Opmerking: in clausule *v)* van definitie 1.6 sluiten we *niet* uit dat $k = 0$; we vatten de definitie dan zo op, dat als $H : \mathbb{N}^2 \to \mathbb{N}$ primitief recursief is en $n \in \mathbb{N}$, de functie F, gedefinieerd door

$$\begin{array}{rcl} F(0) & = & n \\ F(y + 1) & = & H(y, F(y)) \end{array}$$

ook primitief recursief is.

Met een k-plaatsige *relatie* bedoelen we een deelverzameling van $\mathbb{N}^k$. We hebben de volgende conventie voor de *karakteristieke functie* χ_A van een relatie $A \subseteq \mathbb{N}^k$: $\chi_A(\vec{x}) = \begin{cases} 0 & \text{als } \vec{x} \in A \\ 1 & \text{anders} \end{cases}$
Een relatie heet primitief recursief als zijn karakteristieke functie het is.

Opgave 2. Bewijs dat alle primitief recursieve functies berekenbaar zijn. (Hint: realiseer je wat het betekent dat de klasse van primitief recursieve functies wordt *voortgebracht* door de clausules in definitie 1.6, en gebruik proposities 1.3 en 1.4)

Voorbeelden van primitief recursieve functies. De volgende afleidingen laten van een paar eenvoudige functies zien dat ze primitief recursief zijn:

a) $\lambda xy.x + y$ want $0 + y = y = \Pi^1_1(y)$, en
$(x + 1) + y = S(x + y) = S(\Pi^3_2(x, x + y, y))$, dus gedefinieerd met primitieve recursie uit Π^1_1 en de compositie van S en Π^3_2;

b) $\lambda xy.xy$ want $0y = 0 = N(y)$, en
$(x + 1)y = xy + y = (\lambda xy.x + y)(\Pi^3_2(x, xy, y), \Pi^3_3(x, xy, y))$, dus gedefinieerd met primitieve recursie uit N en compositie van $\lambda xy.x + y$ met projecties;

c) $\lambda x.\mathrm{pd}(x)$ waar $\mathrm{pd}(x) = x \dot{-} 1$ want $\mathrm{pd}(0) = 0$, en $\mathrm{pd}(x+1) = x = \Pi^2_1(x, \mathrm{pd}(x))$

Opgave 3. Bewijs dat de volgende functies primitief recursief zijn:

i) $\lambda xy.x^y$

ii) $\lambda xy.x \dot{-} y$

iii) $\lambda xy.\min(x, y)$

iv) sg, waar $\mathrm{sg}(x) = \begin{cases} 1 & \text{als } x > 0 \\ 0 & \text{anders} \end{cases}$

v) $\overline{\mathrm{sg}}$, waar $\overline{\mathrm{sg}}(x) = \begin{cases} 0 & \text{als } x > 0 \\ 1 & \text{anders} \end{cases}$

vi) $\lambda xy.|x - y|$

vii) $\lambda x.n$ voor vaste n

viii) $\lambda x.x!$

ix) $\lambda xy.\mathrm{rm}(x, y)$ waar $\mathrm{rm}(x, y) = 0$ als $y = 0$, en de rest van x bij deling door y, anders.

Opgave 4. Bewijs dat de volgende relaties primitief recursief zijn:

i) $\{(x, y) \mid x = y\}$

ii) $\{(x, y) \mid x \leq y\}$

iii) $\{(x, y) \mid x|y\}$

iv) $\{x \mid x \text{ is een priemgetal}\}$

Opgave 5. Laat zien dat de functie C primitief recursief is, waar

$$C(x, y, z) = \begin{cases} x & \text{als } z = 0 \\ y & \text{anders} \end{cases}$$

Propositie 1.7

a) Als $F : \mathbb{N}^{k+1} \to \mathbb{N}$ primitief recursief is, dan zijn de functies

$$\begin{array}{l} \lambda\vec{x}z.\sum_{y<z} F(\vec{x}, y) \\ \lambda\vec{x}z.\prod_{y<z} F(\vec{x}, y) \\ \lambda\vec{x}z.(\mu y < z.F(\vec{x}, y) = 0) \end{array}$$

het ook (de laatste functie heet gedefinieerd uit F met begrensde minimalisatie, *en geeft de kleinste $y < z$ waarvoor $F(\vec{x}, y) = 0$; als zo'n $y < z$ niet bestaat, geeft hij z);*

b) Als A en B k-plaatsige primitief recursieve relaties zijn, dan zijn $A \cap B$, $A \cup B$, $A \setminus B$ en $\mathbb{N}^k \setminus A$ het ook;

c) Als A een $k+1$-plaatsige primitief recursieve relatie is, dan zijn ook de relaties $\{(\vec{x}, z) \mid \exists y < z(\vec{x}, y) \in A\}$ en $\{(\vec{x}, z) \mid \forall y < z(\vec{x}, y) \in A\}$ primitief recursief.

Bewijs.

a) $\sum_{y<0} F(\vec{x}, y) = 0$ en $\sum_{y<z+1} F(\vec{x}, y) = \sum_{y<z} F(\vec{x}, y) + F(\vec{x}, z)$;
$\prod_{y<0} F(\vec{x}, y) = 1$ en $\prod_{y<z+1} F(\vec{x}, y) = (\prod_{y<z} F(\vec{x}, y))F(\vec{x}, z)$;
$(\mu y < 0.F(\vec{x}, y) = 0) = 0$ en $(\mu y < z + 1.F(\vec{x}, y) = 0) = (\mu y < z.F(\vec{x}, y) = 0) + \mathrm{sg}(\prod_{y<z+1} F(\vec{x}, y))$

b) $\chi_{A\cap B} = \lambda x.\mathrm{sg}(\chi_A(x) + \chi_B(x))$
$\chi_{A\cup B} = \lambda x.\chi_A(x)\chi_B(x)$

Opgave 6. Maak de rest van het bewijs af ■

Propositie 1.8 *Als G_1, G_2 en H n-plaatsige primitief recursieve functies zijn, dan ook de functie F gedefinieerd door*

$$F(\vec{x}) = \begin{cases} G_1(\vec{x}) & \textit{als } H(\vec{x}) = 0 \\ G_2(\vec{x}) & \textit{anders} \end{cases}$$

Bewijs. Immers, $F(\vec{x}) = C(G_1(\vec{x}), G_2(\vec{x}), H(\vec{x}))$, waar C de functie is uit opgave 5, blz. 9. ■

Opgave 7. Laat $p_0, p_1, \ldots$ de rij priemgetallen zijn: $2, 3, 5, \ldots$ Laat zien dat de functie $\lambda n.p_n$ primitief recursief is.

Het is gemakkelijk te zien dat alle primitief recursieve functies *totaal* zijn, dat wil zeggen het domein van een k-plaatsige primitief recursieve functie is de hele $\mathbb{N}^k$. Dus niet elke berekenbare functie is primitief recursief. De vraag ligt nu voor de hand: is misschien elke *totale* berekenbare functie primitief recursief? We zullen in de volgende paragraaf zien, dat het antwoord op deze vraag nee luidt.

1.3 Codering van paren en rijtjes; alternatieve vormen van recursie; totale berekenbare functies die niet primitief recursief zijn

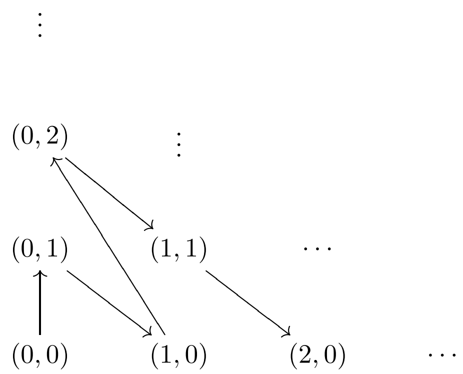
In het vervolg willen we allerlei objecten, zoals programma's, berekeningen enz., opvatten als natuurlijke getallen. Bovendien willen we ons ervan verzekeren dat diverse operaties die op deze objecten kunnen worden toegepast, "primitief recursief" zijn als we ze als getallen opvatten.

We zullen dus *rijtjes* getallen moeten *coderen* als één getal, en laten zien dat allerlei belangrijke operaties op rijtjes, zoals: de lengte van een rijtje nemen, de i-de term bepalen, rijtjes achter elkaar zetten, primitief recursief zijn *in de codes*. Dit wordt hieronder uitgevoerd.

Er zijn vele bijecties: $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$. Zo'n bijectie noemen we een *paringsfunctie*: als $f : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ bijectief is, zeggen we dat $f(x, y)$ *het paar* (x, y) *codeert*. Een voorbeeld van zo'n f is de primitief recursieve functie $\lambda xy.2^x(2y + 1) - 1$.

Opgave 8. Laat $f(x, y) = 2^x(2y + 1) - 1$. Bewijs dat de functies $k_1 : \mathbb{N} \to \mathbb{N}$ en $k_2 : \mathbb{N} \to \mathbb{N}$ die voldoen aan $f(k_1(x), k_2(x)) = x$, primitief recursief zijn.

Een eenvoudiger paringsfunctie wordt gegeven door de "diagonale aftelling" van $\mathbb{N} \times \mathbb{N}$:



Dus $j(0,0)=0,\ j(0,1)=1,\ j(1,0)=2,\ j(0,2)=3$ enz. Er geldt:

$$j(n,m) = \#\{(k,l) \in \mathbb{N}\times\mathbb{N} \mid k+l < n+m \ \vee (k+l = n+m \wedge k < n)\}$$

oftewel

$$j(n,m) = \frac{1}{2}(n+m)(n+m+1)+n = \frac{(n+m)^2+3n+m}{2}$$

De functie j wordt dus gegeven door een polynoom van de graad 2. Er is overigens een stelling (de stelling van Fueter-Pólya, zie het boek van Smoryński) die zegt dat j en zijn 'twist' d.w.z. de functie $\lambda nm.j(m,n)$ de *enige* tweedegraads polynomen zijn die een bijectie: $\mathbb{N}\times\mathbb{N}\to\mathbb{N}$ induceren.

Het is handig dat $x \leq j(x,y)$ en $y \leq j(x,y)$, zodat als we definiëren:

$$\begin{array}{l} j_1(z) = \mu x \leq z.[\exists y \leq z. j(x,y) = z] \\ j_2(z) = \mu y \leq z.[\exists x \leq z. j(x,y) = z] \end{array}$$

dan geldt $j(j_1(z), j_2(z)) = z$.

Opgave 9. Bewijs dit en bewijs dat j_1 en j_2 primitief recursief zijn.

Opgave 10. (Simultane recursie) Stel dat G_1, G_2 : $\mathbb{N}^k \to \mathbb{N}$ en H_1, H_2 : $\mathbb{N}^{k+3} \to \mathbb{N}$ primitief recursief zijn. Definieer de functies F_1 en $F_2 : \mathbb{N}^{k+1} \to \mathbb{N}$ 'simultaan' door het volgende schema:

$$\begin{array}{rclrcl} F_1(0,\vec{x}) & = & G_1(\vec{x}) & F_1(y+1,\vec{x}) & = & H_1(y, F_1(y,\vec{x}), F_2(y,\vec{x}), \vec{x}) \\ F_2(0,\vec{x}) & = & G_2(\vec{x}) & F_2(y+1,\vec{x}) & = & H_2(y, F_1(y,\vec{x}), F_2(y,\vec{x}), \vec{x}) \end{array}$$

Ga na dat F_1 en F_2 goed gedefinieerd zijn, en gebruik de paringsfunctie j en zijn projecties j_1 en j_2 om te laten zien dat F_1 en F_2 primitief recursief zijn.

Nu zijn er in het algemeen n-de graads polynomen P die bijecties $\mathbb{N}^n \to \mathbb{N}$ geven, maar wij gebruiken polynomen van hogere graad:

Definitie 1.9 De bijecties $j^m : \mathbb{N}^m \to \mathbb{N}$ voor $m \geq 1$ zijn gedefinieerd door:

$$\begin{array}{rl} j^1 & \text{is de identitieke functie} \\ j^{m+1}(x_1,\ldots,x_m,x_{m+1}) & = j(j^m(x_1,\ldots,x_m), x_{m+1}) \end{array}$$

We hebben dan ook *projectiefuncties* $j_i^m : \mathbb{N}\to\mathbb{N}$ voor $1 \leq i \leq m$, die voldoen aan

$$j^m(j_1^m(z),\ldots,j_m^m(z)) = z$$

voor alle $z \in \mathbb{N}$, gegeven door:

$$\begin{array}{lcl} j_1^1(z) & = & z \\ j_i^{m+1}(z) & = & \left\{ \begin{array}{rl} j_i^m(j_1(z)) & \text{als } 1 \leq i \leq m \\ j_2(z) & \text{als } i = m+1 \end{array} \right. \end{array}$$

Opgave 11. Bewijs:

i) $j_i^m(j^m(x_1,\ldots,x_m)) = x_i$ voor $1 \leq i \leq m$; en

ii) de functies j^m en j_i^m zijn primitief recursief.

Uit de manier waarop de functies j_i^m met elkaar samenhangen, lijkt het vermoeden gewettigd dat er ook een primitief recursief verband is als we m en i eveneens als variabelen opvatten.

Preciezer geformuleerd:

Propositie 1.10 *De functie F, gedefinieerd door*

$$F(x,y,z) = \left\{ \begin{array}{cl} 0 & \textit{als } y = 0 \textit{ of } y > x \\ j_y^x(z) & \textit{anders} \end{array} \right.$$

is primitief recursief.

Bewijs. Merk eerst op dat de functie $G(w,z) = (j_1)^w(z)$ (de w-de geïtereerde van j_1) primitief recursief is: want $G(0,z) = z$ en $G(w+1,z) = j_1(G(w,z))$. Nu is

$$F(x,y,z) = \begin{cases} 0 & \text{als } y = 0 \text{ of } y > x \\ G(x-1,z) & \text{als } y = 1 \\ j_2(G(x-y,z)) & \text{als } y > 1 \end{cases}$$

Dus F is gedefinieerd uit de primitief recursieve functie G door middel van een (herhaalde) gevalsonderscheiding. ∎

Opgave 12. Werk dit bewijs uit; d.w.z. laat zien dat de in het bewijs gegeven definitie van F correct is, en dat uit deze definitie volgt dat F primitief recursief is.

De functies j^m en hun projecties j_i^m gaven primitief recursieve bijecties: $\mathbb{N}^m \to \mathbb{N}$. Met behulp van propositie 1.10 kunnen we nu een "primitief recursieve" bijectie: $\bigcup_{m\geq 0} \mathbb{N}^m \to \mathbb{N}$ definiëren (het "primitief recursieve" staat tussen aanhalingstekens omdat dit een functie met een variabel aantal argumenten is; voor zulke functies is het begrip primitief recursief natuurlijk niet gedefinieerd). Een element van $\mathbb{N}^m$ voor $m \geq 1$ is een geordend m-tal of *rijtje* $(x_1,\ldots,x_m)$ elementen van $\mathbb{N}$; het enige element van $\mathbb{N}^0$ is het *lege rijtje* (-). Het resultaat van de (te definiëren) functie $\bigcup_{m\geq 0} \mathbb{N}^m \to \mathbb{N}$ op een rijtje $(x_1,\ldots,x_m)$ of (-) noteren we als $\langle x_1,\ldots,x_m\rangle$ of $\langle\,\rangle$ en noemen we de *code van het rijtje*.

Definitie 1.11

$$\begin{array}{rcl} \langle\,\rangle & = & 0 \\ \langle x_0,\ldots,x_{m-1}\rangle & = & j(m-1, j^m(x_0,\ldots,x_{m-1})) + 1 \text{ als } m > 0 \end{array}$$

Opgave 13. Laat zien voor elke $y \in \mathbb{N}$: $y = 0$ of er is een unieke $m > 0$ en unieke $(x_0,\ldots,x_{m-1})$ zodat $y = \langle x_0,\ldots,x_{m-1}\rangle$.

Opmerking. Bij het coderen van rijtjes zijn we begonnen om rijtjes vanaf 0 te indiceren; dit is gebruikelijk in de recursietheorie en ook consistent met de afspraak, de natuurlijke getallen met 0 te laten beginnen.

We hebben nu een paar primitief recursieve functies nodig om effectief te kunnen manipuleren met codes van rijtjes.

Definitie 1.12 De functie $\mathrm{lh}(x)$ geeft ons de *lengte* van het rijtje dat code x heeft, en is gedefinieerd als volgt:

$$\mathrm{lh}(x) = \begin{cases} 0 & \text{als } x = 0 \\ j_1(x-1)+1 & \text{als } x > 0 \end{cases}$$

De functies $(x)_i$ geven ons het i-de element van het rijtje dat code x heeft (vanaf 0 tellen), voor $0 \leq i < \mathrm{lh}(x)$, en 0 anders, en zijn gedefinieerd als

$$(x)_i = \begin{cases} j_{i+1}^{\mathrm{lh}(x)}(j_2(x-1)) & \text{als } x > 0 \text{ en } 0 \leq i < \mathrm{lh}(x) \\ 0 & \text{anders} \end{cases}$$

Opgave 14. Laat zien dat de functies $\lambda x.\mathrm{lh}(x)$ en $\lambda xi.(x)_i$ primitief recursief zijn;
Laat zien dat $(\langle x_0,\ldots,x_{m-1}\rangle)_i = x_i$ en dat $(\langle\,\rangle)_i = 0$;
Laat zien dat voor alle x: $x = 0$ of $x = \langle (x)_0,\ldots,(x)_{\mathrm{lh}(x)-1}\rangle$.

De *concatenatiefunctie* geeft voor twee codes van rijtjes x en y de code van het rijtje dat ontstaat door de rijtjes gecodeerd door x en y achter elkaar te zetten, en wordt genoteerd met $x \star y$. Dus:

$$\begin{array}{rcl} \langle\,\rangle \star y & = & y \\ x \star \langle\,\rangle & = & x \\ \langle (x)_0,\ldots,(x)_{\mathrm{lh}(x)-1}\rangle \star \langle (y)_0,\ldots(y)_{\mathrm{lh}(y)-1}\rangle & = & \langle (x)_0,\ldots(x)_{\mathrm{lh}(x)-1},(y)_0,\ldots,(y)_{\mathrm{lh}(y)-1}\rangle \end{array}$$

Opgave 15. Laat zien dat $\lambda xy.x \star y$ primitief recursief is, door een definitie te geven. (Hint: je kunt eerst een primitief recursieve functie $\lambda xy.x \circ y$ definiëren, zodat:

$$x \circ y = x \star \langle y \rangle$$

Definieer dan, met primitieve recursie, een functie $F(x, y, w)$ door

$$\begin{array}{rcl} F(x, y, 0) & = & x \\ F(x, y, w+1) & = & F(x, y, w) \circ (y)_w \end{array}$$

Zet vervolgens $x \star y = F(x, y, \mathrm{lh}(y))$.)

Recursie naar waardenverloop (Engels: Course-of-values recursion) Het primitief recursieve schema $F(y+1, \vec{x}) = H(y, F(y, \vec{x}), \vec{x})$ stelt ons in staat de waarde van $F(y+1, \vec{x})$ te definiëren direct in termen van $F(y, \vec{x})$. Recursie naar waardenverloop is een schema dat $F(y+1, \vec{x})$ definieert in termen van àlle voorgaande waarden $F(0, \vec{x}), \ldots, F(y, \vec{x})$.

Definitie 1.13 Laten $G : \mathbb{N}^k \to \mathbb{N}$ en $H : \mathbb{N}^{k+2} \to \mathbb{N}$ totale functies zijn. De functie $F : \mathbb{N}^{k+1} \to \mathbb{N}$, gedefinieerd door de clausules

$$\begin{array}{rcl} F(0, \vec{x}) & = & G(\vec{x}) \\ F(y+1, \vec{x}) & = & H(y, \tilde{F}(y, \vec{x}), \vec{x}) \\ (\text{waar } \tilde{F}(y, \vec{x}) & = & j^{y+1}(F(0, \vec{x}), \ldots, F(y, \vec{x})) \end{array}$$

heet gedefinieerd uit G en H door middel van *recursie naar waardenverloop.*

Propositie 1.14 *Stel $G : \mathbb{N}^k \to \mathbb{N}$ en $H : \mathbb{N}^{k+2} \to \mathbb{N}$ zijn primitief recursieve functies en $F : \mathbb{N}^{k+1} \to \mathbb{N}$ is uit G en H gedefinieerd met recursie naar waardenverloop. Dan is ook F primitief recursief.*

Bewijs. Definieer de functie F' door:

$$\begin{array}{rcl} F'(0, \vec{x}) & = & G(\vec{x}) \\ F'(y+1, \vec{x}) & = & j(F'(y, \vec{x}), H(y, F'(y, \vec{x}), \vec{x})) \end{array}$$

Dan is de functie F' duidelijk primitief recursief. Nu volgt met inductie naar y dat $F'(y, \vec{x}) = \tilde{F}(y, \vec{x})$:
$F'(0, \vec{x}) = G(\vec{x}) = j^1(G(\vec{x})) = j^1(F(0, \vec{x})) = \tilde{F}(0, \vec{x})$
$F'(y+1, \vec{x}) = j(F'(y, \vec{x}), H(y, F'(y, \vec{x}), \vec{x})) =$
(wegens inductiehypothese, twee keer gebruikt)
$= j(j^{y+1}(F(0, \vec{x}), \ldots, F(y, \vec{x})), F(y+1, \vec{x})) =$
$= j^{y+2}(F(0, \vec{x}), \ldots, F(y+1, \vec{x}))$.
Er volgt: $F(y, \vec{x}) = \left\{ \begin{array}{rl} G(\vec{x}) & \text{als } y = 0 \\ j_2(F'(y, \vec{x})) & \text{anders} \end{array} \right.$,
dus $F(y, \vec{x}) = C(G(\vec{x}), j_2(F'(y, \vec{x})), y)$ waar C de functie is uit opgave 5, pagina 9. ∎

In plaats van alleen $F(w, \vec{x})$ voor $w \leq y$ te gebruiken (voor de definitie van $F(y+1, \vec{x})$), zouden we er ook aan kunnen denken toe te staan alle waarden $F(w, \vec{x'})$ te gebruiken; dit zou goed gedefinieerd moeten zijn want per inductie hebben we voor alle $w \leq y$ de functies $F_w = \lambda \vec{x}.F(w, \vec{x})$ al gedefinieerd als we met F_{y+1} bezig zijn. Dat dit inderdaad kan, wordt getoond in de volgende opgave:

Opgave 16. Stel $K : \mathbb{N} \to \mathbb{N}$, $G : \mathbb{N}^{k+1} \to \mathbb{N}$ en $H : \mathbb{N}^{k+3} \to \mathbb{N}$ zijn totale functies. Definieer F door:

$$\begin{array}{rcl} F(0, \vec{y}, x) & = & G(\vec{y}, x) \\ F(z+1, \vec{y}, x) & = & H(z, F(z, \vec{y}, K(x)), \vec{y}, x) \end{array}$$

Neem aan dat G, H en K primitief recursief zijn.

a) Bewijs direkt, met behulp van de paringsfunctie j en een aanpassing van de methode van het bewijs van propositie 1.14: als $\forall x(K(x) \le x)$ dan is ook F primitief recursief.

b) Definieer een nieuwe functie F' door:

$$\begin{array}{rcl} F'(0,m,\vec{y},x) & = & G(\vec{y},K^m(x)) \\ F'(n+1,m,\vec{y},x) & = & H(n,F'(n,m,\vec{y},x),\vec{y},K^{m \dot{-} (n+1)}(x)) \end{array}$$

Bewijs: als $n \le m$ dan $\forall k[F'(n,m+k,\vec{y},x) = F'(n,m,\vec{y},K^k(x))]$

c) Bewijs met inductie: $F(z,\vec{y},x) = F'(z,z,\vec{y},x)$ en concludeer dat F primitief recursief is, ook zonder de aanname dat $K(x) \le x$.

Dubbele recursie. Heel anders wordt de zaak, als in de definitie van $F(y+1,\vec{x})$, F_y toegepast wordt op een argument waarin reeds bekende waarden van F_{y+1} mogen voorkomen. We spreken dan van *dubbele recursie*. We behandelen een eenvoudige versie, met een beperkt aantal variabelen.

Definitie 1.15 Laten $G : \mathbb{N} \to \mathbb{N}$, $H : \mathbb{N}^2 \to \mathbb{N}$, $K : \mathbb{N}^4 \to \mathbb{N}$, $J : \mathbb{N} \to \mathbb{N}$, en $L : \mathbb{N}^3 \to \mathbb{N}$ functies zijn; de functie F heet hieruit gedefinieerd *met dubbele recursie* als

$$\begin{array}{rcl} F(0,z) & = & G(z) \\ F(y+1,0) & = & H(y,F(y,J(y))) \\ F(y+1,z+1) & = & K(y,z,F(y+1,z),F(y,L(y,z,F(y+1,z)))) \end{array}$$

Propositie 1.16 *Als G, H, K, J en L primitief recursief zijn en F met dubbele recursie gedefinieerd is als in definitie 1.15 dan zijn alle functies $F_y = \lambda z.F(y,z)$ primitief recursief, maar F zelf hoeft niet primitief recursief te zijn.*

Bewijs. Dat alle F_y primitief recursief zijn volgt uit de definitie. We geven een voorbeeld van een niet-primitief recursieve functie die gedefinieerd kan worden door middel van een soort dubbele recursie. Het idee is om alle definities van primitief recursieve functies: $\mathbb{N} \to \mathbb{N}$ te coderen als getallen; dit gaat als volgt:

- Basisfuncties zijn de functies $\lambda x.0$, $\lambda x.x+1$ en j_i^m, die respectievelijk codes $\langle 0 \rangle$, $\langle 1 \rangle$ en $\langle 2,i,m \rangle$ krijgen;
- als $H, G_1, \ldots, G_p$ codes $n, m_1, \ldots, m_p$ hebben en F is gedefinieerd door

  $$F(x) = H(j^p(G_1(x),\ldots,G_p(x)))$$

  dan heeft F code $\langle 3,n,m_1,\ldots,m_p \rangle$;
- als H en G codes n en m hebben en F is gedefinieerd door

  $$\begin{array}{rcl} F(j(x,0)) & = & G(x) \\ F(j(x,y+1)) & = & H(j^3(x,F(j(x,y)),y)) \end{array}$$

  dan heeft F code $\langle 4,n,m \rangle$.

Ga na, dat iedere éénplaatsige primitief recursieve functie een definitie heeft volgens bovenstaande clausules, en dat dus aan iedere primitief recursieve functie minstens één code wordt toegekend (in feite meer dan één, want de code werkt op de definitie van F, en er kunnen verschillende definities mogelijk zijn).

De volgende stap in het bewijs is nu, een functie Val te definiëren, in feite met een dubbele recursie naar waardenverloop, van twee variabelen k en n, zódat, als k een definitie van de primitief recursieve functie F codeert, $\mathrm{Val}(k,n) = F(n)$. Dit gaat als volgt:

$$\mathrm{Val}(k,x) = \begin{cases} 0 & \text{als } k = \langle 0 \rangle \\ x+1 & \text{als } k = \langle 1 \rangle \\ j_i^m(x) & \text{als } k = \langle 2,i,m \rangle \\ \mathrm{Val}(n, j^p(\mathrm{Val}(m_1,x),\ldots,\mathrm{Val}(m_p,x))) & \text{als} \\ & k = \langle 3,n,m_1,\ldots,m_p \rangle \\ \mathrm{Val}(m, j_1(x)) & \text{als} \\ & k = \langle 4,n,m \rangle \text{ en } j_2(x) = 0 \\ \mathrm{Val}(n, j^3(j_1(x), \mathrm{Val}(k, j(j_1(x), j_2(x)-1)), j_2(x)-1)) & \text{als} \\ & k = \langle 4,n,m \rangle \text{ en } j_2(x) > 0 \\ 0 & \text{anders} \end{cases}$$

Merk op dat $\mathrm{Val}(k,x)$ gedefinieerd wordt in termen van $\mathrm{Val}(n,y)$ met $n < k$ of $n = k \wedge y < x$; dus dit is wel een goed gedefinieerde recursie.

De apotheose van het bewijs is een voorbeeld van *diagonalisatie*, een redenering à la Cantors bewijs van de overaftelbaarheid van $\mathbb{R}$, en een techniek die we nog veel vaker zullen tegenkomen. Stel dat de functie Val primitief recursief is. Dan is ook de functie $\lambda x.\mathrm{Val}(x,x)+1$ primitief recursief; deze functie heeft dan een primitief recursieve definitie gecodeerd door een getal k.
Maar dan geldt per definitie $\mathrm{Val}(k,k) = \mathrm{Val}(k,k)+1$; een tegenspraak is bereikt. ∎

Vergelijken we het schema van dubbele recursie met dat van primitieve recursie, dan zien we dat in de primitief recursieve definitie $F(y+1,x) = H(y, F(y,x), x)$ de functie $F_y = \lambda x.F(y,x)$ wordt toegepast op het argument x, terwijl in het dubbele recursieschema F_y wordt toegepast op een argument dat zelf weer een waarde van F_{y+1} bevat. Dit maakt het mogelijk dat functies gedefinieerd met dubbele recursie zeer hard "groeien". Het eerste voorbeeld van een niet-primitief recursieve functie, gedefinieerd met dubbele recursie uit primitief recursieve functies, werd in 1928 door W. Ackermann gegeven en later door Rosza Péter vereenvoudigd. Een berekenbare totale functie $G(x,y)$ zodat voor elke primitief recursieve functie F er een x_0 is zodat voor alle $x > x_0$, $F(x) < G(x,x)$, heet sindsdien een *Ackermann functie*. Zo'n G kan dus niet primitief recursief zijn (ga na).

Opgave 17.(Ackermann-Péter) Definieer met dubbele recursie:

$$\begin{array}{rcl} A(0,x) & = & x+1 \\ A(n+1,0) & = & A(n,1) \\ A(n+1,x+1) & = & A(n, A(n+1,x)) \end{array}$$

We schrijven weer A_n voor $\lambda x.A(n,x)$. We zeggen voor een primitief recursieve functie $F : \mathbb{N}^k \to \mathbb{N}$ dat F *begrensd is door* A_n, notatie $F \in \mathcal{B}(A_n)$, als voor alle $x_1,\ldots,x_k$ geldt dat $F(x_1,\ldots,x_k) < A_n(x_1+\cdots+x_k)$. Bewijs met inducties op n en x:

i) $n + x < A_n(x)$

ii) $A_n(x) < A_n(x+1)$

iii) $A_n(x) < A_{n+1}(x)$

iv) $A_n(A_{n+1}(x)) \leq A_{n+2}(x)$

v) $nx + 2 \leq A_n(x)$ voor $n \geq 1$

vi) $\lambda x.x+1$, $\lambda x.0$ en $\lambda \vec{x}.x_i \in \mathcal{B}(A_1)$

vii) als $F = \lambda \vec{x}.H(G_1(\vec{x}),\ldots,G_p(\vec{x}))$ en $H, G_1,\ldots,G_p \in \mathcal{B}(A_n)$ voor zekere $n > p$, dan $F \in \mathcal{B}(A_{n+2})$

viii) voor $n \geq 1$: als $F(0,\vec{x}) = G(\vec{x})$ en $F(y+1,\vec{x}) = H(y, F(y,\vec{x}), \vec{x})$ en $G, H \in \mathcal{B}(A_n)$, dan $F \in \mathcal{B}(A_{n+3})$

Concludeer dat voor elke primitief recursieve functie F er een n is met $F \in \mathcal{B}(A_n)$, en dat A een Ackermann functie is.

Opgave 18. Definieer een rij functies $G_0, G_1, \ldots : \mathbb{N} \to \mathbb{N}$ door

$$\begin{array}{rcl} G_0(y) & = & y+1 \\ G_{x+1}(y) & = & (G_x)^{y+1}(y) \end{array}$$

en zet vervolgens $G(x,y) = G_x(y)$. Geef een definitie van G met dubbele recursie en compositie (gebruik een definitieschema voor dubbele recursie waarbij een extra variabele is toegestaan) en bewijs dat G een Ackermann-functie is.

Blijft over, te laten zien dat dubbele recursie wèl altijd berekenbare functies oplevert als de functies die in het schema optreden, berekenbaar zijn. Dit kan wel direkt gedaan worden, maar het is handiger om dit met een beetje meer theorie te doen, die we toch moeten ontwikkelen. We stellen het bewijs dus uit. Ter afsluiting van deze paragraaf nog een paar eenvoudige opgaven.

Opgave 19. Laat zien dat het volgende "recursieschema" geen functie definieert:

$$\begin{array}{rcl} F(0,0) & = & 0 \\ F(x+1,y) & = & F(y,x+1) \\ F(x,y+1) & = & F(x+1,y) \end{array}$$

Opgave 20. Laat zien dat er geen functie is die aan het volgende "recursieschema" voldoet:

$$\begin{array}{rcl} F(0,0) & = & 0 \\ F(x+1,y) & = & F(x,y+1)+1 \\ F(x,y+1) & = & F(x+1,y)+1 \end{array}$$

1.4 Partieel recursieve functies

Notatie. We gaan ons nu weer bezig houden met partiële functies $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$. Omdat voor zulke F de uitdrukking $F(\vec{x})$ niets hoeft te betekenen (voor $\vec{x} \notin \mathrm{dom}(F)$), is het ook niet a priori duidelijk hoe we iets als $F(\vec{x}) = G(\vec{x})$ dienen uit te leggen. We gebruiken daarom in plaats van $=$ het symbool $\simeq$ en we lezen de uitdrukking

$$F(\vec{x}) \simeq G(\vec{x})$$

(en soortgelijke uitdrukkingen) als: de linkerkant is gedefinieerd precies dan als de rechterkant gedefinieerd is, en indien ze gedefinieerd zijn, zijn ze gelijk aan elkaar.

Voorts nemen we aan dat een term pas gedefinieerd kan zijn als al zijn subtermen gedefinieerd zijn. Bijvoorbeeld: de term $\Pi_1^2(x,y)$ lijkt niet van y af te hangen maar toch is $\Pi_1^2(x, G(y))$ pas gedefinieerd als $G(y)$ gedefinieerd is. Vanuit het standpunt van berekenen is dit ook logisch: een machine berekent eerst alle termen van de input, alvorens met de uitkomsten verder te rekenen.

Definitie 1.17 De klasse van *partieel recursieve functies* $\mathbb{N}^k \rightharpoonup \mathbb{N}$ (voor variabele k) wordt voortgebracht door de volgende clausules:

i) alle primitief recursieve functies zijn partieel recursief;

ii) de partieel recursieve functies zijn gesloten onder compositie: als $G_1, \ldots, G_l : \mathbb{N}^k \rightharpoonup \mathbb{N}$ en $H : \mathbb{N}^l \rightharpoonup \mathbb{N}$ partieel recursief zijn, dan ook $\lambda\vec{x}.H(G_1(\vec{x}), \ldots, G_l(\vec{x}))$. Deze functie is gedefinieerd voor alle $\vec{x} \in \bigcap_{i=1}^{l} \mathrm{dom}(G_i)$ waarvoor $(G_1(\vec{x}), \ldots, G_l(\vec{x})) \in \mathrm{dom}(H)$;

iii) als $G : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ partieel recursief is, dan ook F als F gedefinieerd is uit G door middel van minimalisatie (zie propositie 1.5): we schrijven

$$F(\vec{x}) \simeq \mu y.G(\vec{x}, y) = 0$$

$F(\vec{x})$ is gedefinieerd als er een y is zodat $\forall i \leq y.(\vec{x}, i) \in \mathrm{dom}(G)$ en $G(\vec{x}, y) = 0$. $F(\vec{x})$ is dan de kleinste y met die eigenschap.

Definitie 1.18 Een relatie $A \subseteq \mathbb{N}^k$ heet recursief als zijn karakteristieke functie χ_A partieel recursief is.

Een partieel recursieve functie heet *totaal recursief* of kortweg *recursief* als hij totaal is. Omdat χ_A altijd totaal is voor een relatie A, is er geen begrip "partieel recursieve relatie".

Opgave 21. Ga uit van de stelling (die we later zullen bewijzen) dat de partieel recursieve functies precies de berekenbare functies zijn. Laat dan zien dat niet elke recursieve relatie primitief recursief hoeft te zijn.(Hint: gebruik de functie Val uit het bewijs van propositie 1.16)

Propositie 1.19

i) *Als R een recursieve $k+1$-plaatsige relatie is en $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ is gedefinieerd door*

$$F(\vec{x}) \simeq \mu y.R(\vec{x}, y)$$

dan is F partieel recursief;

ii) *Als R een recursieve relatie is en G een partieel recursieve functie, en F is gedefinieerd door*

$$F(x) \simeq \begin{cases} G(x) \text{ als } \exists y.R(y,x) \\ \text{ongedefinieerd anders} \end{cases}$$

dan is F partieel recursief;

Bewijs. Want

i) $F(\vec{x}) \simeq \mu y.\chi_R(\vec{x}, y) = 0$

ii) $F(x) \simeq G((\mu y.\chi_R(y,x) = 0)0 + x)$. Denk aan onze conventie over wanneer uitdrukkingen gedefinieerd zijn!

■

1.5 Codering van registermachine-programma's en de gelijkheid: berekenbaar = partieel recursief; Smn-stelling en recursiestelling

Ons eerste doel in deze paragraaf is te laten zien dat de klassen van berekenbare functies en van partieel recursieve functies samenvallen. We doen dit door een primitief recursieve codering van programma's op te zetten, vervolgens het begrip "berekening" van definitie 1.1 te coderen, zodat we kunnen laten zien:

- De relatie T, gedefinieerd door

  > $T(m, e, x, y)$ geldt precies dan, als y een berekening codeert van de registermachine met programma gecodeerd door e en input $j_1^m(x), \ldots, j_m^m(x)$

  is primitief recursief;

- Er is een primitief recursieve functie U zodat, als $T(m, e, x, y)$, $U(y)$ de uitkomst is van de berekening gecodeerd door y.

Dit is voldoende. Immers stel $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ berekenbaar, dan is er volgens definitie 1.2 een programma P dat F berekent, zeg P heeft code e. Dan volgt weer uit definitie 1.2 dat

$$F(\vec{x}) \simeq U(\mu y.T(k, e, j^k(x_1, \ldots, x_k), y))$$

en dus volgt met definitie 1.17 dat F partieel recursief is. De omgekeerde richting is een gevolg van proposities 1.3–1.5 (ga na).

We zullen dit allemaal hieronder gaan uitvoeren. Dit proces neemt wat tijd, maar bevat geen verrassingen, na de voorbereidingen die we getroffen hebben in de paragraaf over codering van rijtjes.

Coderen we elke basisinstructie van een registermachine-programma:

$$\begin{array}{rcl} r_i^+ \Rightarrow j & \text{als} & \langle i,j\rangle \\ r_i^- \Rightarrow j,k & \text{als} & \langle i,j,k\rangle \end{array}$$

dan kunnen we elk programma P met lijst instructies $(p_1,\ldots,p_n)$ coderen als $\langle\overline{p}_1,\ldots,\overline{p}_n\rangle$, waar $\overline{p}_i$ de code is van instructie p_i (Merk op dat het niet nodig is de instructienummers mee te coderen, aangezien die bepaald worden door de plaats in het rijtje).

Er is nu een primitief recursief predikaat ("predikaat" is een ander woord voor "relatie") Prog zodat $\mathrm{Prog}(e)$ geldt dan en slechts dan als e code is van een programma:

$$\mathrm{Prog}(e) \iff \forall i < \mathrm{lh}(e)(\,((e)_i)_0 \geq 1 \wedge (\mathrm{lh}((e)_i) = 2 \vee \mathrm{lh}((e)_i) = 3))\ \wedge \mathrm{lh}(e) > 0$$

(De extra eis dat $\mathrm{lh}(e) > 0$ m.a.w. dat we lege programma's uitsluiten, heeft te maken met onze formalisering van *berekening* als een rij machinetoestanden waar een machinetoestand een rij registerinhouden is, samen met informatie over welke instructie op dat moment uit te voeren)

De definitie van T is nu een directe omzetting van definitie 1.1. We definiëren $T(m,e,x,y)$ als de conjunctie van de volgende beweringen:

- $\mathrm{Prog}(e)$
- $\mathrm{lh}(y) > 1$
- $\forall i < \mathrm{lh}(y)[\mathrm{lh}((y)_i) = \max\{m+2, \max\{((e)_i)_0 + 2 \mid 0 \leq i < \mathrm{lh}(e)\}\}]$
- $((y)_0)_0 = 1\ \wedge\ ((y)_0)_1 = 0\ \wedge \forall i \leq m(1 \leq i \Rightarrow ((y)_0)_{i+1} = j_i^m(x))$
- $\forall i < \mathrm{lh}((y)_0)[m+1 < i \Rightarrow ((y)_0)_i = 0]$
- $\forall i < \mathrm{lh}(y)[((y)_i)_0 > \mathrm{lh}(e) \Leftrightarrow i = \mathrm{lh}(y) - 1]$
- $\forall k,l < e\,\forall i < \mathrm{lh}(y) - 1[(e)_{((y)_i)_0} = \langle k,l\rangle \Rightarrow$
$$(y)_{i+1} = (y)_i[l/((y)_i)_0, ((y)_i)_k + 1/((y)_i)_k]$$
- $\forall k,l,m < e\,\forall i < \mathrm{lh}(y) - 1[(e)_{((y)_i)_0} = \langle k,l,m\rangle \Rightarrow$
$$\{((y)_i)_k = 0\ \wedge\ (y)_{i+1} = (y)_i[m/((y)_i)_0]\} \vee$$
$$\{((y)_i)_k > 0\ \wedge\ (y)_{i+1} = (y)_i[l/((y)_i)_0, ((y)_i)_k - 1/((y)_i)_k]\}]$$

Hierbij staat $y[a/(y)_i]$ voor de code van het rijtje dat ontstaat uit y door zijn i-de component door a te vervangen; $y[a/(y)_i, b/(y)_j]$ staat voor het uitvoeren van twee vervangingen. Ga na dat deze "substitutie"-operatie primitief recursief is in y, a, i. Door bovenstaande lijst rustig door te lezen overtuig je je ervan dat $T(m,e,x,y)$ inderdaad de bedoelde betekenis heeft. Verder is $T(m,e,x,y)$ een conjunctie van primitief recursieve predicaten, dus primitief recursief.

Aangezien de uitkomst van een berekening het getal is dat aan het eind in het eerste register staat, geeft

$$U(y) = ((y)_{\mathrm{lh}(y)-1})_1$$

inderdaad de uitkomst van de door y gecodeerde berekening (àls y een berekening codeert), en de functie U is duidelijk primitief recursief.

De notaties T en U zijn standaard in de recursietheorie. Het predikaat T heet ook wel het *Kleene T-predikaat*, en U heet de *uitkomstfunctie*. Merk op dat we eigenlijk iets sterkers hebben bewezen dan louter dat elke berekenbare functie partieel recursief is:

Propositie 1.20 *Definieer de partieel recursieve functie* Φ *door*

$$\Phi(m,e,x) \simeq U(\mu y.T(m,e,x,y))$$

Dan is er voor elke partieel recursieve functie $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ *een* $e \in \mathbb{N}$ *zodat*

$$F(x_1,\ldots,x_k) \simeq \Phi(k,e,j^k(x_1,\ldots,x_k))$$

voor alle $x_1,\ldots,x_k$*; met andere woorden er is een partieel recursieve opsomming van alle partieel recursieve functies.*

De functie Φ uit propositie 1.20 is partieel recursief en dus berekenbaar door een programma; zo'n programma heet een *universeel* programma.

Ter contrast een propositie die laat zien dat de analoge uitspraak voor *totaal* recursieve functies niet opgaat:

Propositie 1.21 *Er is géén totaal recursieve functie* $\Psi(m,e,x)$ *zodat elke totaal recursieve functie* $F : \mathbb{N}^m \to \mathbb{N}$ *gelijk is aan*

$$\lambda x_1 \cdots x_m.\Psi(m,e,j^m(x_1,\ldots,x_m))$$

voor zekere e*.*

Bewijs. Want stel wel, dan was ook de functie

$$\lambda x_1 \cdots x_m.\Psi(m,j^m(x_1,\ldots,x_m),j^m(x_1,\ldots,x_m))+1$$

totaal recursief, dus gelijk aan $\lambda x_1 \cdots x_m.\Psi(m,e,j^m(x_1,\ldots,x_m))$ voor zekere e; maar voor die e zou dan gelden

$$\Psi(m,e,e) = \Psi(m,e,e)+1$$

tegenspraak (vergelijk het bewijs van propositie 1.16) ∎

Opgave 22. Laat zien dat er voor elke m-plaatsige partieel recursieve functie F oneindig veel getallen e zijn zodat

$$F = \lambda x_1 \cdots x_m.\Phi(m,e,j^m(x_1,\ldots,x_m))$$

Als $F = \lambda x_1 \cdots x_m.\Phi(m,e,j^m(x_1,\ldots,x_m))$ noemen we e ook wel een code voor F, en we schrijven vaak

$$e \bullet (x_1,\ldots,x_m)$$

voor $\Phi(m,e,j^m(x_1,\ldots,x_m))$.

De volgende belangrijke stelling luistert naar de merkwaardige naam "S^m_n-stelling". Een betere benaming, die je ook wel hoort, is "parametriseringsstelling". Hij zegt dat codes van partieel recursieve functies die van parameters afhangen, primitief recursief in de parameters te verkrijgen zijn. Het bewijs is eenvoudig.

Stelling 1.22 (S^m_n-stelling; Kleene) *Voor elke* $m \geq 1$ *en* $n \geq 1$ *is er een* $m+1$*-plaatsige primitief recursieve functie* S^m_n *zodat voor alle* $e,x_1,\ldots,x_m,y_1,\ldots,y_n$*:*

$$S^m_n(e,x_1,\ldots,x_m) \bullet (y_1,\ldots,y_n) \simeq e \bullet (x_1,\ldots,x_m,y_1,\ldots,y_n)$$

Bewijs. Als $\neg\mathrm{Prog}(e)$ zetten we $S^m_n(e,x_1,\ldots,x_m) = e$. Als e een programma P codeert, zodat

$$\begin{array}{l} 0\,a_1 \cdots a_m\,b_1 \cdots b_n \\ \Downarrow\ P \\ e \bullet (a_1,\ldots,a_m,b_1,\ldots,b_n)\,a_1 \cdots a_m\,b_1 \cdots b_n\,\vec{c} \end{array}$$

is het de bedoeling dat voor alle $x_1, \ldots, x_m$, $S^m_n(e, x_1, \ldots, x_m)$ het volgende programma codeert:

$$\begin{array}{ll}
 & 0\, b_1 \cdots b_n \\
\text{registerinhouden } \vec{b} \text{ opschuiven} & \Downarrow\ K_{2,n+m+2} \cdots K_{n+1,2n+m+1} \\
 & 0\, \vec{0}\, b_1 \cdots b_n \\
\text{registerinhouden } \vec{x} \text{ inbrengen} & \Downarrow\ (R_2^+)^{x_1} \cdots (R_{m+1}^+)^{x_m} \\
 & 0\, x_1 \cdots x_m\, \vec{0}\, b_1 \cdots b_n \\
\text{aansluiten} & \Downarrow\ K_{n+m+2,m+2} \cdots K_{2n+m+1,m+n+1} \\
 & 0\, x_1 \cdots x_m\, b_1 \cdots b_n \\
 & \Downarrow\ P \\
 & e \bullet (x_1, \ldots, x_m, b_1, \ldots b_n)\, x_1 \cdots x_m\, b_1 \cdots b_n\, \vec{c}
\end{array}$$

($(R_i^+)^n$ staat voor het programma dat n keer 1 optelt bij het i-de register) Het is duidelijk dat een code voor dit programma primitief recursief te verkrijgen is in $e, x_1, \ldots, x_m$. ∎

Opgave 23. Laat zien dat de functie S^m_n *stijgend* is in $x_1, \ldots, x_m$.

Gevolg 1.23 *Er is een primitief recursieve functie H zodat voor alle e, f, x:*

$$H(e, f) \bullet x \simeq e \bullet (f \bullet x)$$

Bewijs. De functie $\lambda efx.e \bullet (f \bullet x)$ is partieel recursief; immers

$$e \bullet (f \bullet x) \simeq U(j_2(\mu z.[T(1, f, x, j_1(z)) \wedge T(1, e, U(j_1(z)), j_2(z))]))$$

Dus $e \bullet (f \bullet x) \simeq g \bullet (e, f, x)$ voor zekere code g; zet $H(e, f) = S^2_1(g, e, f)$ ∎

Een ander gevolg van de S^m_n-stelling doet in het begin enigszins bizar aan, omdat het ons in staat stelt (codes van) partieel recursieve functies F te definiëren in termen van de te definiëren code zèlf! (natuurlijk is dit niet echt zo. Omdat elke partieel recursieve functie oneindig veel codes heeft, wordt er niets "gedefinieerd". Toch heeft dit gevolg iets "mysterieus")

Gevolg 1.24 (Recursiestelling, Kleene 1938, eerste versie) *Voor elke partieel recursieve functie $F : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ met $k \geq 1$ is er een code e zodat voor alle $x_1, \ldots, x_k$:*

$$e \bullet (x_1, \ldots x_k) \simeq F(x_1, \ldots, x_k, e)$$

Bewijs. Laat f een code zijn voor F, zodat $f \bullet (x_1, \ldots, x_{k+1}) \simeq F(x_1, \ldots, x_{k+1})$ voor alle $x_1, \ldots, x_{k+1}$. Laat nu g een code zijn zodat voor alle $h, y, x_1, \ldots, x_k$:

$$g \bullet (h, y, x_1, \ldots, x_k) \simeq h \bullet (x_1, \ldots, x_k, S^1_k(y, y))$$

Laat nu

$$e = S^1_k(S^1_{k+1}(g, f), S^1_{k+1}(g, f))$$

Dan geldt

$$\begin{array}{ccl}
e \bullet (x_1, \ldots, x_k) & \simeq & \\
S^1_k(S^1_{k+1}(g, f), S^1_{k+1}(g, f)) \bullet (x_1, \ldots, x_k) & \simeq & \text{met } S^m_n\text{-stelling} \\
S^1_{k+1}(g, f) \bullet (S^1_{k+1}(g, f), x_1, \ldots, x_k) & \simeq & \\
g \bullet (f, S^1_{k+1}(g, f), x_1, \ldots, x_k) & \simeq & \text{per definitie van } g \\
f \bullet (x_1, \ldots, x_k, S^1_k(S^1_{k+1}(g, f), S^1_{k+1}(g, f))) & \simeq & \text{per definitie van } e \\
f \bullet (x_1, \ldots, x_k, e) & \simeq & \text{per definitie van } f \\
F(x_1, \ldots, x_k, e) & &
\end{array}$$

∎

Opgave 24. Laten $R_1, \ldots, R_n \subseteq \mathbb{N}^k$ recursieve relaties zijn zodat $R_i \cap R_j = \emptyset$ voor $i \neq j$; stel $G_1, \ldots, G_n : \mathbb{N}^k \rightharpoonup \mathbb{N}$ partieel recursieve functies. Dan is de partiële functie F, gedefinieerd door

$$F(\vec{x}) \simeq \begin{cases} G_1(\vec{x}) & \text{als } R_1(\vec{x}) \\ \vdots & \vdots \\ G_n(\vec{x}) & \text{als } R_n(\vec{x}) \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

ook partieel recursief. Bewijs dit.

Gevolg 1.25 *De klasse van partieel recursieve functies is gesloten onder primitieve recursie: als $G : \mathbb{N}^k \rightharpoonup \mathbb{N}$ en $H : \mathbb{N}^{k+2} \rightharpoonup \mathbb{N}$ partieel recursief zijn en F is uit G en H gedefinieerd met primitieve recursie, dan is ook F partieel recursief.*

Bewijs. Laten g en h codes voor resp. G en H zijn. Volgens de recursiestelling is er een code f zodat voor alle $y, \vec{x}$:

$$f \bullet (y, \vec{x}) \quad \simeq \quad \begin{cases} g \bullet (\vec{x}) & \text{als } y = 0 \\ h \bullet (y-1, f \bullet (y-1, \vec{x}), \vec{x}) & \text{als } y > 0 \end{cases}$$

Ga na dat $f \bullet (y, \vec{x}) \simeq F(y, \vec{x})$ voor alle $y, \vec{x}$. ■

Gevolg 1.26 *De recursieve relaties zijn gesloten onder begrensde kwantoren: als $R \subseteq \mathbb{N}^{k+1}$ recursief is, zijn ook*

$$\{(\vec{x}, y) \mid \forall w < y. R(\vec{x}, w)\}$$

en

$$\{(\vec{x}, y) \mid \exists w < y. R(\vec{x}, w)\}$$

recursief.

Bewijs. Immers, de karakteristieke functies van deze relaties worden gedefinieerd met primitieve recursie uit de karakteristieke functie van R. ■

Inspecteren we nog eens het bewijs van de recursiestelling, dan zien we dat de daar gedefinieerde e in feite een primitief recursieve functie is van de code f van F. Hieruit volgt:

1) Er is een primitief recursieve functie G_n zodat voor alle $x_1, \ldots, x_n, f$:

$$G_n(f) \bullet (x_1, \ldots, x_n) \simeq f \bullet (x_1, \ldots, x_n, G_n(f))$$

2) Omdat de functies S_n^m stijgend zijn in alle argumenten, i.h.b. injectief, en elke partieel recursieve functie oneindig veel codes heeft, zijn er voor elke F oneindig veel e als in de recursiestelling.

Opgave 25. Laat zien dat de volgende keuzes van codes voor partieel recursieve functies gedaan kunnen worden:

i) Gegeven een recursief predikaat R, kies e zodat voor alle $\vec{x}$:

$$e \bullet (\vec{x}) \quad \simeq \quad \begin{cases} 0 & \text{als } R(\vec{x}, e) \\ 1 & \text{anders} \end{cases}$$

ii) Gegeven een recursief predikaat R en een partieel recursieve functie F, kies e zodat voor alle $\vec{x}$:

$$e \bullet (\vec{x}) \quad \simeq \quad \begin{cases} F(\vec{x}) & \text{als } \exists y. R(\vec{x}, y, e) \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

De tweede versie van de recursiestelling is een direkt gevolg van de eerste.

Gevolg 1.27 (recursiestelling, tweede versie) *Voor elke partieel recursieve functie F en elke n is er een code e zodat voor alle $x_1, \ldots, x_n$:*

$$e \bullet (x_1, \ldots, x_n) \simeq F(e) \bullet (x_1, \ldots, x_n)$$

Bewijs. Laat $G(x_1, \ldots, x_n, e) \simeq F(e) \bullet (x_1, \ldots, x_n)$ en pas de eerste versie van de recursiestelling toe om een e te vinden zodat $e \bullet (x_1, \ldots, x_n) \simeq G(x_1, \ldots, x_n, e)$. ∎

Opgave 26. Bewijs de primitief recursieve versie. Dus: er is een primitief recursieve functie T zodat voor alle f en alle $x_1, \ldots, x_n$:

$$T(f) \bullet (x_1, \ldots, x_n) \simeq (f \bullet T(f)) \bullet (x_1, \ldots, x_n)$$

Opgave 27. Bewijs de *recursiestelling met parameters*: er is een primitief recursieve F zodat voor alle $f, \vec{y}, \vec{x}$:

$$F(f, \vec{y}) \bullet (\vec{x}) \simeq f \bullet (F(f, \vec{y}), \vec{y}, \vec{x})$$

alsmede: er is een primitief recursieve F' zodat voor alle $f, \vec{y}, \vec{x}$:

$$F'(f, \vec{y}) \bullet (\vec{x}) \simeq (f \bullet (F'(f, \vec{y}), \vec{y})) \bullet (\vec{x})$$

Notatie. De partieel recursieve functie $\lambda x.\Phi(1, e, x)$ wordt vaak aangegeven met φ_e

Opmerking. De tweede versie van de recursiestelling wordt wel eens "dekpuntsstelling" genoemd; geven we met φ_e de functie aan met code e, dan is φ_e "dekpunt" van de "operatie" $\varphi_e \mapsto \varphi_{F(e)}$. Dit is natuurlijk maar bij wijze van spreken zo.

Immers, er ìs geen operatie van het gegeven type, omdat F niet op functies, maar op codes werkt (het kan heel wel zijn, dat φ_e dezelfde functie is als $\varphi_{e'}$ terwijl toch $\varphi_{F(e)} \neq \varphi_{F(e')}$!)

Als besluit van dit hoofdstuk passen we de recursiestelling toe, om het beloofde bewijs te geven dat de klasse van totale berekenbare functies gesloten is onder definitie met dubbele recursie. Stel dus dat G, H, J, K en L totaal recursief zijn en F gedefinieerd is door:

$$\begin{array}{rcl} F(0, z) & = & G(z) \\ F(y+1, 0) & = & H(y, F(y, J(y))) \\ F(y+1, z+1) & = & K(y, z, F(y+1, z), F(y, L(y, z, F(y+1, z)))) \end{array}$$

Dan is ook F totaal recursief; immers we kunnen met de recursiestelling een code f kiezen zodat

$$f\bullet(y, z) \simeq \left\{ \begin{array}{rl} G(z) & \text{als } y = 0 \\ H(y-1, f \bullet (y-1, J(y-1))) & \text{als } y > 0 \\ & \text{en } z = 0 \\ K(y-1, z-1, f \bullet (y, z-1), f \bullet (y-1, L(y-1, z-1, f \bullet (y, z-1)))) & \text{als } y > 0 \\ & \text{en } z > 0 \end{array} \right.$$

Opgave 28. Bewijs met dubbele inductie (op y en z) dat de boven gedefinieerde functie φ_f totaal is, en gelijk aan F.

Nog een laatste opgave.
Opgave 29. Bewijs de *dubbele recursiestelling van Smullyan*: gegeven 2-plaatsige partieel recursieve functies F en G zijn er voor iedere k codes a en b zodat voor alle $x_1, \ldots, x_k$:

$$a \bullet (x_1, \ldots, x_k) \simeq F(a, b) \bullet (x_1, \ldots, x_k)$$

en

$$b \bullet (x_1, \ldots, x_k) \simeq G(a, b) \bullet (x_1, \ldots, x_k)$$

2 Onbeslisbaarheid en recursief opsombare verzamelingen

2.1 Oplosbare problemen

In dit hoofdstuk vatten we relaties vaak op als *problemen*. Het "probleem" is dan om uit te maken, of een bepaald k-tal een element van de relatie is of niet. Bijvoorbeeld: de relatie

$$\{(f, x) \mid \exists z T(1, f, x, z)\}$$

zien we als het probleem: is $f \bullet x$ gedefinieerd?

We noemen een probleem *oplosbaar* als de bijbehorende relatie recursief is; met andere woorden als er een totale berekenbare functie bestaat die voor alle waarden van de parameters het antwoord op het probleem geeft. Een ander woord is *beslisbaar*.

Het probleem: is $f \bullet x$ gedefinieerd? is klassiek en heet het *stop-probleem* (stopt de berekening van het programma met code f, voor input x?), in het Engels *Halting Problem*.

Propositie 2.1 *Het stop-probleem is niet oplosbaar.*

Bewijs. Stel F is een totaal recursieve functie zodat voor alle f, x:

$$F(f, x) \simeq \begin{cases} 0 & \text{als } f \bullet x \text{ gedefinieerd} \\ 1 & \text{anders} \end{cases}$$

Als G een partieel recursieve functie is zodat $\mathrm{dom}(G) = \mathbb{N} \setminus \{0\}$ (bijvoorbeeld $G(x) \simeq \mu z.xz > 1$), dan is er met de recursiestelling een code f zodat voor alle x:

$$f \bullet x \simeq G(F(f, x))$$

Duidelijk geldt: $f \bullet x$ gedefinieerd $\iff$ $F(f, x) \neq 0$ $\iff$ $f \bullet x$ ongedefinieerd; tegenspraak. ∎

Het kan eenvoudiger. Het *standaard-probleem* is: is $x \bullet x$ gedefinieerd? Onoplosbaarheid van het standaardprobleem impliceert natuurlijk onoplosbaarheid van het stop-probleem (ga na). Stel het standaard-probleem is oplosbaar via een recursieve functie F, zodat voor alle x geldt:

$$F(x) \simeq \begin{cases} 0 & \text{als } x \bullet x \text{ gedefinieerd} \\ 1 & \text{anders} \end{cases}$$

Kies weer G als in het bewijs van propositie 2.1, en zij f en code voor de partieel recursieve functie $\lambda x.G(F(x))$. Dan geldt $f \bullet f$ ongedefinieerd $\iff$ $F(f) = 0$ $\iff$ $f \bullet f$ gedefinieerd; tegenspraak. Dus de recursiestelling is niet echt nodig in het bewijs van propositie 2.1.

De verhouding van het standaard-probleem tot het stop-probleem is een voorbeeld van *reduceerbaarheid* van het ene probleem tot het andere: als $R \subseteq \mathbb{N}^m$ en $S \subseteq \mathbb{N}^k$ heet R *reduceerbaar* tot S als er totaal recursieve functies $F_1, \ldots, F_k : \mathbb{N}^m \to \mathbb{N}$ bestaan zodat voor alle $\vec{x} \in \mathbb{N}^m$: $\vec{x} \in R \Leftrightarrow (F_1(\vec{x}), \ldots, F_k(\vec{x})) \in S$.

Opgave 30. Laat zien: als R reduceerbaar tot S en S is oplosbaar, dan is ook R oplosbaar.

Later zullen we deze reduceerbaarheid nog meer in detail tegenkomen. Een veralgemenisering van het begrip "oplosbaar" is het begrip "oplosbaar met betrekking tot…" Het probleem is dan om uit te maken of $\vec{x} \in R$, gegeven dat $\vec{x} \in P$, en een oplossing is een partieel recursieve functie $F : \mathbb{N}^m \rightharpoonup \mathbb{N}$ zodat voor alle $\vec{x}$:

$$\vec{x} \in P \Rightarrow [F(\vec{x}) \text{ gedefinieerd en } F(\vec{x}) = \begin{cases} 0 & \text{als } \vec{x} \in R \\ 1 & \text{anders} \end{cases}]$$

Voorbeeld. Het probleem: geldt $0 \in \mathrm{rge}(\varphi_e)$, gegeven dat φ_e totaal is, is niet oplosbaar (We schrijven $\mathrm{rge}(F)$ voor het *bereik* van F).

Bewijs. We moeten laten zien dat er geen partieel recursieve F bestaat zodat voor alle e: als φ_e totaal is, is $F(e)$ gedefinieerd, en er geldt:

$$F(e) = \begin{cases} 0 & \text{als er } z \text{ is met } e \bullet z = 0 \\ 1 & \text{anders} \end{cases}$$

Laat g een code zijn zodat voor alle x, y:

$$g \bullet (x, y) \quad \simeq \quad \begin{cases} 0 & \text{als } T(1, x, x, y) \\ 1 & \text{anders} \end{cases}$$

Dan is $S^1_1(g, x)$ de code van een totale functie, want $S^1_1(g, x) \bullet y \simeq g \bullet (x, y)$. Er geldt:

$$0 \in \mathrm{rge}(\varphi_{S^1_1(g,x)}) \Leftrightarrow \exists y.T(1, x, x, y) \Leftrightarrow x \bullet x \text{ is gedefinieerd}$$

Dus als F bestond, dan was de functie $G = \lambda x.F(S^1_1(g, x))$ een oplossing van het standaardprobleem; een tegenspraak met propositie 2.1. ∎

Opgave 31. Bewijs dat het probleem: is φ_e totaal?, niet oplosbaar is (Hint: definieer een totaal recursieve F zodat voor alle e: $e \bullet e$ is gedefinieerd $\Longleftrightarrow$ $\varphi_{F(e)}$ is totaal; en concludeer hieruit wat je moet bewijzen).

Opgave 32. Bewijs dat de volgende problemen onbeslisbaar zijn:

i) $\mathrm{dom}(\varphi_e) = \emptyset$?

ii) $\mathrm{rge}(\varphi_e)$ oneindig?

iii) $\varphi_e = \varphi_f$?

Opgave 33. Bepaal welke van de volgende problemen beslisbaar zijn voor codes e van totale functies φ_e:

i) $\exists x.e \bullet x \neq 0$

ii) $\exists x.e \bullet x \leq e \bullet (x + 1)$

iii) $\exists x.e \bullet x \geq e \bullet (x + 1)$

iv) $e \bullet x = y$ (als 3-plaatsige relatie)

Opgave 34. Laat zien dat er geen totaal recursieve $F : \mathbb{N}^2 \to \mathbb{N}$ is zodat voor alle e, x geldt:

$$e \bullet x \text{ gedefinieerd} \implies \exists y(T(1, e, x, y) \wedge \mathrm{lh}(y) \leq F(e, x))$$

(Hint: laat zien dat er een primitief recursieve functie $S(e, x, n)$ bestaat zodat $S(e, x, n)$ de eerste n stappen van een berekening van programma e met input x geeft, mits uiteraard $\mathrm{Prog}(e)$; concludeer dan uit het bestaan van een F als boven dat het stop-probleem beslisbaar is)

Opgave 35. Laat zien dat er geen totaal recursieve $F : \mathbb{N} \to \mathbb{N}$ is zodat voor alle e: als φ_e de karakteristieke functie van een eindige verzameling is, dan $\forall n(e \bullet n = 0 \to n \leq F(e))$
(Hint: beschouw de verzameling $\{\mathrm{lh}(\mu y.T(1, e, x, y))\}$)

In opgave 32 worden eigenschappen van codes genoemd die eigenlijk eigenschappen van de gecodeerde partiële functies zijn: d.w.z. als e en f dezelfde partiële functie coderen, dan heeft e de eigenschap dan en slechts dan als f de eigenschap heeft. Zulke eigenschappen noemen we *extensioneel* (hebben alleen te maken met hoe φ_e zich als partiële functie "uitstrekt"):

Definitie 2.2 Een deelverzameling X van $\mathbb{N}$ heet *extensioneel* als voor alle $e, f \in \mathbb{N}$ geldt:

$$\text{als } \varphi_e = \varphi_f \text{ en } e \in X \text{ dan ook } f \in X$$

De *Stelling van Rice* zegt, dat niet-triviale extensionele eigenschappen van natuurlijke getallen nooit beslisbaar kunnen zijn.

Stelling 2.3 (Rice) *Als $X \subseteq \mathbb{N}$ recursief en extensioneel is, dan $X = \emptyset$ of $X = \mathbb{N}$.*

Bewijs. Ga zelf na, dat als X recursief en extensioneel is, het complement van X het ook is; als $X \neq \emptyset$, $X \neq \mathbb{N}$ mogen we dus wel aannemen dat alle codes voor de lege functie niet in X zitten. Laat f zo'n code zijn, en kies $e \in X$.
Met de S^m_n-stelling is er een primitief recursieve functie F zodat voor alle x, y:

$$F(x) \bullet y \ \simeq \ \begin{cases} e \bullet y & \text{als } \exists z T(1, x, x, z) \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

Dan geldt:
$x \bullet x$ gedefinieerd $\iff \ \varphi_{F(x)} = \varphi_e \ \Rightarrow \ F(x) \in X$
$x \bullet x$ ongedefinieerd $\iff \ \varphi_{F(x)} = \varphi_f \ \Rightarrow F(x) \notin X$
Dus $x \bullet x$ gedefinieerd $\Leftrightarrow$ $F(x) \in X$ voor alle $x \in \mathbb{N}$; m.a.w. het standaard-probleem is reduceerbaar tot X. Maar dan kan X niet recursief zijn. ∎

Voor wie de recursiestelling leuk vindt, een alternatief bewijs van stelling 2.3: stel X recursief, extensioneel, $e \in X$, $f \notin X$ met f een code van de lege functie. Met de recursiestelling is er een code g zodat voor alle x:

$$g \bullet x \ \simeq \ \begin{cases} f \bullet x & \text{als } g \in X \\ e \bullet x & \text{anders} \end{cases}$$

Dan volgt omdat X extensioneel is:

$$g \in X \ \Rightarrow \ \varphi_g = \varphi_f \ \Rightarrow g \notin X$$

en

$$g \notin X \ \Rightarrow \ \varphi_g = \varphi_e \ \Rightarrow \ g \in X$$

zodat een tegenspraak is bereikt.

2.2 Recursief opsombare verzamelingen

Eén graadje ingewikkelder van structuur dan de recursieve verzamelingen zijn de *recursief opsombare* verzamelingen. Als R recursief opsombaar is, is er in het algemeen een partieel recursieve functie F zodat R precies de verzameling nulpunten van F is; echter voor $x \notin R$ hoeft $F(x)$ niet gedefinieerd te zijn.

Propositie 2.4 *De volgende vier uitspraken zijn equivalent voor een deelverzameling $R \subseteq \mathbb{N}$:*

i) Voor zekere partieel recursieve functie $F : \mathbb{N} \rightharpoonup \mathbb{N}$ is $R = \mathrm{dom}(F)$;

ii) Voor zekere recursieve relatie $S \subseteq \mathbb{N}^2$ is $R = \{x \mid \exists y((x, y) \in S)\}$;

iii) Voor zekere partieel recursieve $F : \mathbb{N} \rightharpoonup \mathbb{N}$ is $R = \mathrm{rge}(F)$;

iv) $R = \emptyset$ of $R = \mathrm{rge}(F)$ voor zekere primitief recursieve functie F.

Bovendien is de equivalentie i)⇔ii)⇔iii) primitief recursief in codes*, d.w.z. bijvoorbeeld voor i)⇒ii): er is een primitief recursieve functie P zodat voor alle f, $P(f)$ de code van een karakteristieke functie is, en* $\mathrm{dom}(\varphi_f) = \{x \mid \exists y.P(f) \bullet (x, y) = 0\}$*, etc.*

Bewijs. i)⇒ii). Laat f een code voor F. Dan is $\mathrm{dom}(F) = \{x \mid \exists y T(1, f, x, y)\}$; omdat T primitief recursief is, is er een code g zodat voor alle f, x, y:

$$g \bullet (f, x, y) \ \simeq \ \begin{cases} 0 & \text{als } T(1, f, x, y) \\ 1 & \text{anders} \end{cases}$$

Dan is $\mathrm{dom}(\varphi_f) = \{x \mid \exists y(S^1_2(g,f) \bullet (x,y) = 0)\}$, en $\varphi_{S^1_2(g,f)}$ is altijd een code van een karakteristieke functie.
ii)⇒iii). We kunnen een code g kiezen, zodat als f de code van een karakteristieke functie F is, en $R = \{x \mid \exists y(f \bullet (x,y) = 0)\}$, voor alle x:

$$g \bullet (f,x) \simeq \begin{cases} x & \text{als } \exists y(f \bullet (x,y) = 0) \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

Dan is $R = \mathrm{rge}(\varphi_{S^1_1(g,f)})$ (ga na).
iii)⇒i). Als $R = \mathrm{rge}(\varphi_f)$ dan $R = \mathrm{dom}(\varphi_{S^1_1(g,f)})$, waar g een code is zodat voor alle f, x:

$$g \bullet (f,x) \simeq \begin{cases} 0 & \text{als } \exists y[T(1,f,j_1(y),j_2(y)) \wedge U(j_2(y)) = x] \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

iv)⇒iii) is duidelijk.
iii)⇒iv). Als $R = \mathrm{rge}(F)$ dan is hetzij $R = \emptyset$, hetzij $a \in R$ voor zekere a. In het eerste geval zijn we klaar; in het andere geval, zij f een code voor F, en laat g een code zijn zodat

$$g \bullet (f,y) \simeq \begin{cases} a & \text{als } \neg T(1,f,j_1(y),j_2(y)) \\ U(j_2(y)) & \text{als } T(1,f,j_1(y),j_2(y)) \end{cases}$$

Dan is $R = \mathrm{rge}(\varphi_{S^1_1(g,f)})$, en $\varphi_{S^1_1(g,f)}$ is duidelijk primitief recursief. ∎

De equivalentie van iv) met de andere uitspraken kan niet primitief recursief in codes worden gemaakt omdat we de gevalsonderscheiding: is $\mathrm{rge}(\varphi_f) = \emptyset$? niet recursief kunnen doen (dit is een onoplosbaar probleem).

Definitie 2.5 Een deelverzameling A van $\mathbb{N}$ heet *recursief opsombaar* (Engels: recursively enumerable, we korten af: r.e.) als hij aan een van de alternatieven van propositie 2.4 voldoet. We zeggen ook wel dat A een *r.e. verzameling* is.
Natuurlijk kunnen we dit uitbreiden tot deelverzamelingen van $\mathbb{N}^k$: noem $A \subseteq \mathbb{N}^k$ recursief opsombaar of r.e., als $\{x \in \mathbb{N} \mid (j^k_1(x), \dots, j^k_k(x)) \in A\}$ het is (Equivalent: $A \subseteq \mathbb{N}^k$ is r.e. als er een k-plaatsige partieel recursieve functie φ_e is zodat $A = \mathrm{dom}(\varphi_e)$).
We gebruiken de volgende codes voor recursief opsombare verzamelingen: met de definitie

$$W^k_e = \mathrm{dom}(\varphi_e) = \{(x_1, \dots, x_k) \mid \exists z T(k, e, j^k(x_1, \dots, x_k), z)\}$$

is elke r.e. verzameling A gelijk aan zekere W^k_e. (Als $k = 1$ schrijven we W_e in plaats van W^1_e) We noemen e de *code van A als r.e. verzameling*.

Voorbeelden.

i) De onoplosbaarheid van het standaardprobleem betekent dat de verzameling

$$\mathcal{K} = \{x \mid x \bullet x \text{ is gedefinieerd}\}$$

niet recursief is; $\mathcal{K}$ is wel r.e. want $\mathcal{K} = \{x \mid \exists y.T(1,x,x,y)\}$ dus voldoet aan ii) van propositie 2.4 (de letter $\mathcal{K}$ is standaard voor deze verzameling).

ii) Elke recursieve verzameling is r.e. (ga na).

Opgave 36. Bewijs: als S r.e. is en R is reduceerbaar tot S, dan is R r.e.

Opgave 37. Bewijs dat $\{e \mid \varphi_e \text{ is totaal}\}$ niet r.e. is.

Opgave 38. Bewijs dat elke r.e. verzameling van de vorm $\{x \mid \exists y((x,y) \in S)\}$ is, voor zekere *primitief* recursieve $S \subseteq \mathbb{N}^2$.

Propositie 2.6

i) $R \subseteq \mathbb{N}^k$ *is recursief* $\Longleftrightarrow$ R *is r.e. en het complement van* R *is r.e.;*

ii) $\mathcal{K}^c = \{x \mid x \bullet x$ *is ongedefinieerd*$\}$ *is niet r.e.;*

iii) $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ *is partieel recursief* $\Longleftrightarrow$ *de grafiek van* F : $\{(\vec{x}, y) \mid F(\vec{x}) = y\}$ *is r.e.;*

iv) als $F : \mathbb{N}^k \to \mathbb{N}$ *totaal recursief is, is zijn grafiek recursief*

Bewijs.

i) Als R recursief is, is zijn complement het ook, dus zijn beide r.e.; als $R = \{x \mid \exists y S(x, y)\}$ en $R^c = \{x \mid \exists y.T(x, y)\}$ definieer dan $F(x) \simeq \mu y.(S(x, y) \vee T(x, y))$. F is totaal en er geldt: $x \in R \Leftrightarrow S(x, F(x))$.

ii) Dit volgt uit i) want als $\mathcal{K}^c$ r.e. was, was $\mathcal{K}$ recursief.

iii) Zij f een code voor F. De grafiek van F is

$$\{(\vec{x}, y) \mid y = F(\vec{x})\} = \{(\vec{x}, y) \mid \exists w[T(k, f, j^k(\vec{x}), w) \wedge U(w) = y]\}$$

en is dus r.e.; omgekeerd, als

$$\{(\vec{x}, y) \mid y = F(\vec{x})\} = \{(\vec{x}, y) \mid \exists z S(\vec{x}, y, z)\}$$

voor zekere recursieve S, dan is $F = \varphi_g$, waar g een code is zodat

$$\forall \vec{x}[g \bullet (\vec{x}) \simeq j_1(\mu z.S(\vec{x}, j_1(z), j_2(z)))]$$

iv) Laat G_F de grafiek van F; G_F is r.e. wegens iii) maar omdat F totaal is, is

$$(G_F)^c = \{(\vec{x}, y) \mid \exists z[T(k, f, j^k(\vec{x}), z) \wedge U(z) \neq y]\}$$

voor een code f van F; dus óók r.e. Dus G_F is recursief vanwege i).

■

Opgave 39. Bewijs dat i) van propositie 2.6 primitief recursief in codes geldt, d.w.z. als $W_x = W_y^c$ dan $\chi_{W_x} = \varphi_{T(x,y)}$ voor zekere primitief recursieve T.

Opgave 40.

i) Gebruik de recursiestelling om te laten zien dat voor elke totaal recursieve F er een n is zodat $W_{F(n)} = W_n$; leid hieruit af dat er een n is zodat $W_n = \{n\}$ en een n zodat $W_n = \{m \mid m > n\}$.

ii) Bewijs, met de recursiestelling, dat er een totaal recursieve F is zodat $\forall n(F(n) < F(n+1))$ en $\forall n(W_{F(n)} = \{F(n+1)\})$.
(Hint: gebruik, dat de S_n^m-functies *stijgend* zijn)

Opgave 41. Bewijs:

i) Als F totaal recursief en niet-dalend is, en een oneindig bereik heeft, dan is rge(F) recursief.

ii) Elke oneindige r.e. verzameling bevat een oneindige recursieve deelverzameling.

Gelden deze resultaten primitief recursief in codes?

Uit propositie 2.6ii) zien we dat de r.e. verzamelingen niet gesloten zijn onder complement nemen. Uit opgave 37 zien we dat ze ook niet gesloten zijn onder willekeurige doorsneden (want $\{e \mid \varphi_e$ is totaal$\} = \bigcap_{x \in \mathbb{N}} \{e \mid \exists y T(1, e, x, y)\}$) of verenigingen (want $A = \bigcup_{x \in A} \{x\}$ voor elke verzameling A; en singletons zijn natuurlijk r.e.). Wel geldt:

Propositie 2.7

i) Als R en S r.e. zijn, dan zijn $R \cup S$ en $R \cap S$ het ook;

ii) Als $R \subseteq \mathbb{N}^{k+1}$ r.e. is, dan ook

$$\{(\vec{x}, z) \in \mathbb{N}^{k+1} \mid \forall w < z R(\vec{x}, w)\}$$

iii) Als $R \subseteq \mathbb{N}$ r.e. is, dan is $\bigcup_{x \in R} W_x$ het ook.

Deze resultaten gelden ook primitief recursief in codes (d.w.z. i): $W_x \cap W_y = W_{F(x,y)}$ en $W_x \cup W_y = W_{G(x,y)}$ voor primitief recursieve F en G, etc.).

Opgave 42. Bewijs deze propositie. Voor ii): gebruik codering van rijtjes om een kwantorcombinatie $\forall w < z \exists y$ te vervangen door een combinatie $\exists y \forall w < z$.

We beëindigen deze paragraaf met twee resultaatjes die nogal eens toepassing vinden: de *reductiestelling* en het bestaan van *disjuncte, recursief onscheidbare r.e. verzamelingen.*

Propositie 2.8 (Reductiestelling) *Voor elk tweetal r.e. verzamelingen $X, Y \subseteq \mathbb{N}$ bestaan er r.e. X' en $Y' \subseteq \mathbb{N}$ zodat:*

i) $X' \subseteq X$ en $Y' \subseteq Y$;

ii) $X' \cap Y' = \emptyset$;

iii) $X' \cup Y' = X \cup Y$.

Bewijs. Stel $X = \{x \mid \exists y R(x, y)\}$ en $Y = \{x \mid \exists y S(x, y)\}$ voor recursieve $R, S \subseteq \mathbb{N}^2$. Laat

$$X' = \{x \mid \exists y[R(x, y) \wedge \forall w \leq y \neg S(x, w)]\}$$

$$Y' = \{x \mid \exists y[S(x, y) \wedge \forall w < y \neg R(x, w)]\}$$

■

Definitie 2.9 *R.e. verzamelingen $X, Y \subseteq \mathbb{N}$ heten* recursief onscheidbaar *als er geen recursieve $R \subseteq \mathbb{N}$ is met $X \subseteq R$ en $R \cap Y = \emptyset$ (equivalent: als er geen totaal recursieve functie $f : \mathbb{N} \to \mathbb{N}$ is met $f(x) = 0$ voor alle $x \in X$ en $f(x) > 0$ voor alle $x \in Y$).*

Propositie 2.10 *Er bestaan disjuncte, recursief onscheidbare r.e. verzamelingen $X, Y \subseteq \mathbb{N}$.*

Bewijs. Zet

$$X = \{x \mid x \bullet x = 0\} = \{x \mid \exists y[T(1, x, x, y) \wedge U(y) = 0]\}$$

$$Y = \{x \mid x \bullet x = 1\}$$

We laten nu zien: er is een primitief recursieve functie F zodat voor alle e en f: als $X \subseteq W_e$, $Y \subseteq W_f$ en $W_e \cap W_f = \emptyset$, dan $F(e, f) \notin W_e \cup W_f$ (Ga na, dat hieruit de bewering volgt! Gebruik propositie 2.6i)).
Zij g een code zodat voor alle e, f, x:

$$g \bullet (e, f, x) \simeq \begin{cases} 1 & \text{als } \exists y[T(1, e, x, y) \wedge \forall w \leq y \neg T(1, f, x, w)] \\ 0 & \text{als } \exists y[T(1, f, x, y) \wedge \forall w < y \neg T(1, e, x, w)] \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

Dan geldt voor $F(e, f) = S^2_1(g, e, f)$ als e en f als boven:
als $S^2_1(g, e, f) \in W_e$ dan

$$S^2_1(g, e, f) \bullet S^2_1(g, e, f) = g \bullet (e, f, S^2_1(g, e, f)) = 1$$

dus $S^2_1(g,e,f) \in Y$, tegenspraak;
als $S^2_1(g,e,f) \in W_f$ dan

$$S^2_1(g,e,f) \bullet S^2_1(g,e,f) = 0$$

dus $S^2_1(g,e,f) \in X$; eveneens een tegenspraak.
Dus $S^2_1(g,e,f) \notin W_e \cup W_f$. ∎

Opgave 43. (Het "extensieprobleem"). Laat zien dat er partieel recursieve functies G bestaan die niet zijn voort te zetten tot een totaal recursieve functie (m.a.w. er is geen totaal recursieve functie F zodat F en G samenvallen op het domein van G).

(Hint: je kunt propositie 2.10 gebruiken, maar nodig is dit niet)

Opgave 44. Een r.e. verzameling X heet *creatief* via een partieel recursieve functie F als voor alle e geldt:

$$W_e \cap X = \emptyset \implies F(e) \text{ gedefinieerd en } F(e) \notin X \cup W_e$$

Laat zien: als X en Y de verzamelingen uit het bewijs van 2.10 zijn, dan zijn er functies F en G zodat X en Y resp. creatief via F en G zijn.

2.3 Extensionele r.e. verzamelingen en effectieve operaties: openheid, monotonie en continuïteit

In deze paragraaf verzamel ik drie belangrijke stellingen over extensionele verzamelingen van (partieel) recursieve functies en r.e. verzamelingen, en "effectieve" operaties op deze dingen. Eerst maar even duidelijk maken wat we bedoelen.

Definitie 2.11 Een deelverzameling $A \subseteq \mathbb{N}$ heet *extensioneel als verzameling codes van partieel recursieve functies* als er een verzameling F van partiële functies $\mathbb{N} \rightharpoonup \mathbb{N}$ is zodat $A = \{e \mid \varphi_e \in F\}$. Geheel analoog is het begrip: A is extensioneel als verzameling codes van totaal recursieve functies. $A \subseteq \mathbb{N}$ heet *extensioneel als verzameling codes van r.e. verzamelingen* als er een verzameling V van r.e. verzamelingen is zodat $A = \{e \mid W_e \in V\}$. Schrijf $\mathcal{PR}$ voor de collectie van partieel recursieve functies, $\mathcal{R}$ voor de verzameling totaal recursieve functies, en $\mathcal{RE}$ voor de verzameling van r.e. deelverzamelingen van $\mathbb{N}$. Een functie $H : \mathcal{PR} \to \mathcal{PR}$, $\mathcal{R} \to \mathcal{R}$ of $\mathcal{RE} \to \mathcal{RE}$ heet een *effectieve operatie* als er een recursieve functie F bestaat zodat voor alle e : $H(\varphi_e) = \varphi_{F(e)}$ resp. $H(W_e) = W_{F(e)}$. In het geval van een operatie $\mathcal{R} \to \mathcal{R}$ hoeft F slechts gedefinieerd te zijn op codes van totaal recursieve functies.

De bedoeling in deze paragraaf is nu, aan te tonen dat er zeer natuurlijke topologieën op $\mathcal{PR}$, $\mathcal{R}$, $\mathcal{RE}$ zijn, zodat geldt:

- elke r.e. extensionele verzameling codes is *open* (althans in het geval van $\mathcal{PR}$ en $\mathcal{RE}$);
- elke effectieve operatie is *continu*.

Ten bate van diegenen, die onbekend zijn met sommige van bovenstaande begrippen, volgt eerst een mini-stoomcursus topologie.

2.3.1 Enkele basisbegrippen uit de topologie

Definitie 2.12 Een *topologische ruimte* is een verzameling X met daarbij gegeven een *topologie*, dat is een collectie $\mathcal{U}$ van deelverzamelingen van X, die aan de volgende drie eisen voldoet:

1. als U_1 en $U_2 \in \mathcal{U}$ dan $U_1 \cap U_2 \in \mathcal{U}$
2. als $\{U_i \mid i \in I\} \subseteq \mathcal{U}$ dan $\bigcup_{i \in I} U_i \in \mathcal{U}$
3. $X \in \mathcal{U}$.

We noemen de elementen van $\mathcal{U}$ *open verzamelingen*.

Voorbeelden:

- $\mathbb{R}$ met de "gewone" (of Euclidische) topologie, d.w.z. $U \subseteq \mathbb{R}$ is open $\iff$ voor iedere $x \in U$ is er een $\epsilon > 0$ zodat het open interval $(x-\epsilon, x+\epsilon)$ bevat is in U;
- Een verzameling X met $\mathcal{U} = \mathcal{P}(X)$, de machtsverzameling van X: iedere deelverzameling van X is open (equivalent: $\{x\}$ is open voor iedere $x \in X$). Dit noemen we de *discrete* topologie;
- Een rol in het vervolg zal spelen de *Sierpinski-ruimte*, dit is de verzameling $2 = \{0,1\}$ met 3 open verzamelingen: $\emptyset$, $\{0\}$, $\{0,1\}$.
- Gegeven een topologische ruimte $(X,\mathcal{U})$ en een deelverzameling $A \subseteq X$ kunnen we een topologie $\mathcal{U}_A$ op A definiëren door te zetten $\mathcal{U}_A = \{U \cap A \mid U \in \mathcal{U}\}$. Deze topologie heet de *deelruimte-* of *relatieftopologie*.

Elementen $x \in X$ heten ook wel *punten* van de topologische ruimte $(X,\mathcal{U})$. In de topologie zijn punten eigenlijk van ondergeschikt belang en spelen open verzamelingen de hoofdrol. Als $x \in U \in \mathcal{U}$ is het goed om over U te denken als een "benadering van x". Zo kan een open intervalletje rond een reëel getal x beschouwd worden als een eindige benadering van x.

Definitie 2.13 Zij $(X,\mathcal{U})$ een topologische ruimte. Een *basis* voor $\mathcal{U}$ is een deelverzameling $\mathcal{B} \subseteq \mathcal{U}$ waarvoor geldt:

1. als $B_1, B_2 \in \mathcal{B}$ en $x \in B_1 \cap B_2$ is er $B_3 \in \mathcal{B}$ met $x \in B_3 \subseteq B_1 \cap B_2$, en
2. Voor elke $x \in X$ en $x \in U \in \mathcal{U}$ is er $B \in \mathcal{B}$ met $x \in B \subseteq U$.

Gegeven een basis $\mathcal{B}$, is hieruit $\mathcal{U}$ volledig bepaald; immers

$$U \in \mathcal{U} \Leftrightarrow \exists \mathcal{C} \subseteq \mathcal{B} \text{ met } U = \bigcup \mathcal{C}$$

We zeggen dat $\mathcal{U}$ wordt *voortgebracht* door $\mathcal{B}$. $\mathcal{B}$ heet *een basis* als 1. van de vorige definitie geldt, en tevens

$$X = \bigcup \mathcal{B}$$

Voorbeeld. De collectie open intervallen $\{(p,q) \mid p,q \in \mathbb{Q}\}$ is een basis voor de topologie op $\mathbb{R}$. De collectie singletons $\{x\}$ is een basis voor de discrete topologie op X.

Definitie 2.14 Laten $(X,\mathcal{U})$, $(Y,\mathcal{V})$ topologische ruimten zijn. Een afbeelding $f : X \to Y$ heet *continu* (voor de gegeven topologieën) als voor elke $x \in X$ en voor elke benadering $V \in \mathcal{V}$ van $f(x)$ (i.e. $f(x) \in V$) er een benadering U van x is zodat $f[U] \subseteq V$. Equivalent hiermee is: als voor elke $V \in \mathcal{V}$, $f^{-1}(V) \in \mathcal{U}$

Ga na, dat de samenstelling van twee continue afbeeldingen weer continu is.

Opgave 45. Laten $\mathcal{B}_1$ en $\mathcal{B}_2$ resp. bases voor topologieën op X en Y zijn. Bewijs dat $f : X \to Y$ continu is dan en slechts dan als voor iedere $x \in X$ en iedere $B \in \mathcal{B}_2$ met $f(x) \in B$ er een $B' \in \mathcal{B}_1$ is met $x \in B'$ en $f[B'] \subseteq B$

Produkt-topologieën. Gegeven een collectie topologische ruimten $\{(X_i,\mathcal{U}_i) \mid i \in I\}$ definiëren we een topologie op de produktverzameling $\prod_{i\in I} X_i = \{(x_i)_{i\in I} \mid \forall i \in I. x_i \in X_i\}$ als volgt: laat, voor $j \in I$ en $U_j \in \mathcal{U}_j$,

$$\pi_j^{-1}(U_j) = \{(x_i)_{i\in I} \mid x_j \in U_j\}$$

De collectie

$$\{\bigcap_{n=1}^{k} \pi_{i_n}^{-1}(U_{i_n}) \mid k \in \mathbb{N}, i_1, \ldots, i_k \in I, U_{i_n} \in \mathcal{U}_{i_n} \text{ voor } 1 \leq n \leq k\}$$

is dan een basis voor een topologie op $\prod_{i\in I} X_i$ (ga na!). We noemen deze topologie de *produkt-topologie* op $\prod_{i\in I} X_i$.

Opgave 46.

1. Laat zien dat de projecties $\pi_i : \prod_{i \in I} X_i \to X_i$ continu zijn met betrekking tot de produkt-topologie;

2. Gegeven een collectie continue afbeeldingen

$$\{Y \xrightarrow{f_i} X_i \mid i \in I\},$$

definieer $f : Y \to \prod_{i \in I} X_i$ door

$$f(y) = (f_i(y))_{i \in I}$$

Laat zien dat f continu is.

De collectie totale functies $\mathbb{N} \to \mathbb{N}$ kunnen we zien als een $\mathbb{N}$-voudig produkt van kopieën van $\mathbb{N}$:

$$\mathbb{N}^{\mathbb{N}} = \prod_{n \in \mathbb{N}} \mathbb{N} = \{(x_n)_{n \in \mathbb{N}} \mid x_n \in \mathbb{N}\}$$

Geven we $\mathbb{N}$ de discrete topologie en vervolgens $\mathbb{N}^{\mathbb{N}}$ de produkttopologie, dan wordt een basis voor deze topologie gegeven door de verzamelingen

$$\mathcal{O}_\sigma = \{f \in \mathbb{N}^{\mathbb{N}} \mid \forall i < \mathrm{lh}(\sigma) f(i) = (\sigma)_i\}$$

Opgave 47. Bewijs dit.

Evenzo kunnen we $\mathcal{P}(\mathbb{N})$ zien als $2^{\mathbb{N}} = \prod_{n \in \mathbb{N}} 2$. Geven we $2 = \{0, 1\}$ de discrete topologie, dan is een basis voor de produkt-topologie gegeven door de verzamelingen

$$\mathcal{O}_{U,V} = \{A \subseteq \mathbb{N} \mid U \subseteq A \wedge V \cap A = \emptyset\}$$

voor U en V *eindige* deelverzamelingen van $\mathbb{N}$.

Opgave 48. Bewijs dit.

We vatten nu $\mathcal{R}$, de verzameling totaal recursieve functies, op als topologische ruimte: $\mathcal{R} \subseteq \mathbb{N}^{\mathbb{N}}$ dus we geven $\mathcal{R}$ de relatieftopologie.
Op dezelfde manier zouden we $\mathcal{RE}$ de relatieftopologie van $\mathcal{P}(\mathbb{N})$ kunnen geven, maar vanuit recursietheoretisch standpunt is dit niet de "juiste" topologie. Immers als $A = W_e$ is *positieve* informatie over $x \in A$ altijd te verkrijgen mits $x \in A$: we laten het programma met code e werken op x. Echter als $x \notin A$ zullen we dit op deze manier nooit weten. De goede benaderingen van r.e. verzamelingen zouden moeten zijn: verzamelingen

$$\mathcal{O}_U = \{A \subseteq \mathbb{N} \mid U \subseteq A\}$$

voor $U \subseteq \mathbb{N}$ *eindig.*

Opgave 49. Bewijs, dat de verzamelingen $\mathcal{O}_U$ een basis vormen voor de topologie op $2^{\mathbb{N}}$ die we krijgen door 2 de Sierpinski-topologie te geven, en vervolgens $2^{\mathbb{N}}$ de produkt-topologie.

We vatten dus $\mathcal{RE}$ op als deelruimte (met de relatieftopologie) van het $\mathbb{N}$-voudig produkt van de Sierpinski-ruimte.
Een soortgelijk probleem als met $\mathcal{RE}$ hebben we met $\mathcal{PR}$: ten eerste vormen partiële functies niet op een natuurlijke manier een deelverzameling van de totale functies; ten tweede, als $\varphi_e(x)$ ongedefinieerd is, zullen we dit nooit weten. We doen daarom het volgende:
Partiële functies $\mathbb{N} \rightharpoonup \mathbb{N}$ kunnen we zien als totale functies $\mathbb{N} \to \tilde{N}$, waar $\tilde{N} = \mathbb{N} \cup \{\perp\}$, $\perp$ is een of ander symbool waarvan we aannemen $\perp \notin \mathbb{N}$. Immers voor $f : \mathbb{N} \rightharpoonup \mathbb{N}$ definiëren we $\overline{f} : \mathbb{N} \to \tilde{N}$ door $\overline{f}(x) = \perp$ als $f(x)$ ongedefinieerd is, en $\overline{f}(x) = f(x)$ voor $x \in \mathrm{dom}(f)$. We geven $\tilde{N}$ de topologie $\{\tilde{N}\} \cup \mathcal{P}(\mathbb{N})$ (dus de enige benadering van $\perp$ is $\tilde{N}$ zelf). Vervolgens geven we $\prod_{n \in \mathbb{N}} \tilde{N}$ de produkt-topologie, en $\mathcal{PR} \subseteq \prod_{n \in \mathbb{N}} \tilde{N}$ de relatieftopologie.

Opgave 50. Bewijs, dat de aldus gedefinieerde topologie op $\mathcal{PR}$ voortgebracht wordt door de basis

$$\{\mathcal{O}_a \mid a : \mathbb{N} \rightharpoonup \mathbb{N}, \mathrm{dom}(a) \textit{ eindig}\}$$

met

$$\mathcal{O}_a = \{f \in \mathcal{PR} \mid \mathrm{dom}(a) \subseteq \mathrm{dom}(f) \wedge \forall x \in \mathrm{dom}(a)\, a(x) = f(x)\}$$

Opgave 51. De Sierpinski-ruimte en $\tilde{N}$ zijn eigenlijk twee voorbeelden van één constructie van topologische ruimten. Verduidelijk.

De volgende propositie, en de opgaven daarna, geven criteria voor openheid van deelverzamelingen van $\mathcal{PR}$ en $\mathcal{RE}$, en voor continuïteit van functies $\mathcal{PR} \to \mathcal{PR}$ en $\mathcal{RE} \to \mathcal{RE}$, bijna geheel in termen van de partiële ordening $\subseteq$:

Propositie 2.15 *Schrijf, voor partiële functies $f, g : \mathbb{N} \rightharpoonup \mathbb{N}$, $f \subseteq g$ als de grafiek van f bevat is in de grafiek van g. Een functie $F : \mathcal{PR} \to \mathcal{PR}$ is continu dan en slechts dan als F aan de volgende twee voorwaarden voldoet:*

1. *F is* monotoon*, i.e. $f \subseteq g \Rightarrow F(f) \subseteq F(g)$*
2. *F is* compact*, i.e. voor elke f en elke* eindige *$a \subseteq F(f)$ is er eindige $b \subseteq f$ zodat $a \subseteq F(b)$*

Bewijs. Stel F continu, $f \subseteq g$, $a \subseteq F(f)$ eindig. Dan is $F(f)$ een element van de basis-open verzameling $\mathcal{O}_a$, dus per definitie van continuïteit is er een basiselement $\mathcal{O}_b$ met $f \in \mathcal{O}_b$ en $F[\mathcal{O}_b] \subseteq \mathcal{O}_a$, d.w.z. $\forall h(b \subseteq h \Rightarrow a \subseteq F(h))$. Dit bewijst meteen 2. Wegens $f \subseteq g$ volgt nu $a \subseteq F(g)$; bijgevolg geldt voor iedere eindige a: $a \subseteq F(f) \Rightarrow a \subseteq F(g)$. Maar dan moet gelden $F(f) \subseteq F(g)$; i.e. 1.
De omgekeerde implicatie is eenvoudig. ∎

Opgave 52. Bewijs dezelfde bewering voor $\mathcal{RE}$, d.w.z. $F : \mathcal{RE} \to \mathcal{RE}$ is continu dan en slechts dan als

1. $A \subseteq B \Rightarrow F(A) \subseteq F(B)$ voor r.e. verzamelingen A en B; en
2. Voor iedere A en iedere eindige $U \subseteq F(A)$ is er eindige $V \subseteq A$ met $U \subseteq F(V)$

beide gelden.

Opgave 53. Bewijs voor deelverzamelingen X van $\mathcal{PR}$ of $\mathcal{RE}$:
X is open dan en slechts dan als

1. $A \subseteq B \wedge A \in X \Rightarrow B \in X$ (X is naar boven gesloten), en
2. $A \in X \Rightarrow$ er is eindige $U \subseteq A$ met $U \in X$

beide gelden.

2.3.2 Stellingen van Myhill-Shepherdson, Rice-Shapiro en Kreisel-Lacombe-Shoenfield

Stelling 2.16 (Myhill-Shepherdson)

1. *Laat $R \subseteq \mathbb{N}$ r.e. en extensioneel als verzameling codes van partieel recursieve functies. Dan is $\{\varphi_e \mid e \in R\}$ open in $\mathcal{PR}$.*
2. *Zij $F : \mathcal{PR} \to \mathcal{PR}$ een effectieve operatie. Dan is F continu.*

Bewijs. We gebruiken de criteria gegeven in propositie 2.15 en opgave 53.
1. Stel $R = W_h$. Laat G de partieel recursieve functie zijn, gedefinieerd door:

$$G(g, e, f, x) \simeq \mu z.[T(1, e, x, z) \vee (T(1, h, S^2_1(g, e, f), j_1(z)) \wedge T(1, f, x, j_2(z)))]$$

Kies nu met de recursiestelling een code g zodat

$$g \bullet (e,f,x) \quad \simeq \quad \left\{ \begin{array}{rl} U(G(g,e,f,x)) & \text{als } T(1,e,x,G(g,e,f,x)) \\ U(j_2(G(g,e,f,x))) & \text{als } \neg T(1,e,x,G(g,e,f,x)) \\ & \wedge\ T(1,f,x,j_2(G(g,e,f,x))) \end{array} \right.$$

Stel nu $e \in R$, $\varphi_e \subseteq \varphi_f$. Als $S^2_1(g,e,f) \notin R$ dan volgt (met de definitie van G en de keuze van g) dat $\varphi_{S^2_1(g,e,f)} = \varphi_e \Rightarrow S^2_1(g,e,f) \in R$ (wegens R extensioneel); tegenspraak. Bijgevolg $S^2_1(g,e,f) \in R$ en dus (omdat $\varphi_e \subseteq \varphi_f$) $\varphi_{S^2_1(g,e,f)} = \varphi_f$, waardoor volgt $f \in R$. Dus $\{\varphi_e \mid e \in R\}$ is naar boven gesloten.
Vervolgens moeten we laten zien dat als $R = W_h$ en $e \in R$, er een u is met φ_u eindig, $\varphi_u \subseteq \varphi_e$ en $u \in R$. Kies, weer met de recursiestelling, een code g' zodat voor alle e, x:

$$g' \bullet (e,x) \quad \simeq \quad \left\{ \begin{array}{rl} e \bullet x & \text{als } \neg\exists z \leq x\, T(1,h,S^1_1(g',e),z) \\ \text{ongedefinieerd} & \text{anders} \end{array} \right.$$

Stel nu $e \in R$. Als $S^1_1(g',e) \notin R$ volgt $\varphi_{S^1_1(g',e)} = \varphi_e$ dus $S^1_1(g',e) \in R$, tegenspraak; dus $S^1_1(g',e) \in R$. Maar nu is $\varphi_{S^1_1(g',e)}$ eindig, want voor $x \geq \mu w.T(1,h,S^1_1(g',e),w)$ geldt dat $S^1_1(g',e) \bullet x \simeq g' \bullet (e,x)$ ongedefinieerd is. Tevens geldt $\varphi_{S^1_1(g',e)} \subseteq \varphi_e$, zodat conditie 2. van opgave 53 ook vervuld is. Dit bewijst dat $\{\varphi_e \mid e \in R\}$ open is in $\mathcal{PR}$.
2. Het is genoeg, te laten zien dat $F^{-1}(\mathcal{O}_a)$ open is in $\mathcal{PR}$ voor elke basis-open verzameling $\mathcal{O}_a = \{f \in \mathcal{PR} \mid a \subseteq f\}$ met $a : \mathbb{N} \rightharpoonup \mathbb{N}$ eindig. Gegeven zo'n eindige a, is er een (code van een) rijtje σ zodat $\mathrm{dom}(a) = \{i < \mathrm{lh}(\sigma) \mid (\sigma)_i > 0\}$ en zodat voor alle $i \in \mathrm{dom}(a) :\ a(i) = (\sigma)_i - 1$. Omdat F een effectieve operatie is, is er een totaal recursieve functie ψ zodat $F(\varphi_e) = \varphi_{\psi(e)}$ voor alle e; dus

$$F^{-1}(\mathcal{O}_a) = \{\varphi_e \mid \forall i < \mathrm{lh}(\sigma)((\sigma)_i > 0 \Rightarrow \exists y.(T(1,\psi(e),i,y) \wedge U(y) = (\sigma)_i - 1))\}$$

Ga nu na dat $\{e \mid \varphi_e \in F^{-1}(\mathcal{O}_a)\}$ r.e. en extensioneel is, zodat $F^{-1}(\mathcal{O}_a)$ open is met deel 1. van het bewijs. ∎

Merk op, dat we eigenlijk een sterkere bewering bewezen hebben (in 1. van stelling 2.16) dan alleen dat $\{\varphi_e \mid e \in R\}$ open is: we lieten zien dat er een *primitief recursieve* functie S is, zodat voor alle e: als $e \in R$ dan $S(e) \in R$, $\varphi_{S(e)}$ eindig en $\varphi_{S(e)} \subseteq \varphi_e$. Een dergelijke versterking is ook mogelijk voor deel 2.

Opgave 54. Laat, voor eindige rijtjes σ,

$$\mathcal{O}_\sigma = \{f \mid \forall i < \mathrm{lh}(\sigma)((\sigma)_i > 0 \Rightarrow F(i) = (\sigma)_i - 1)\}$$

Laat zien (in de situatie van deel 2. van 2.16): er is een primitief recursieve functie S zodat voor alle f en σ: als $F(\varphi_f) \in \mathcal{O}_\sigma$ dan is $\varphi_{S(f,\sigma)} \subseteq \varphi_f$, $\varphi_{S(f,\sigma)}$ eindig, en $F(\varphi_{S(f,\sigma)}) \in \mathcal{O}_\sigma$.

Opgave 55. Concludeer uit de stelling van Myhill-Shepherdson dat er geen totaal recursieve functie F is zodat voor alle e geldt: φ_e is constant (mogelijk partieel) dan en slechts dan als $F(e) \in \mathcal{K}$.

De stelling van Rice-Shapiro is het analogon van Myhill-Shepherdson voor $\mathcal{RE}$.

Stelling 2.17 (Rice-Shapiro, 1955) *1. Laat $X \subseteq \mathbb{N}$ een r.e. extensionele verzameling codes voor r.e. verzamelingen zijn. Dan is $\{W_e \mid e \in X\}$ open in $\mathcal{RE}$.*

2. Zij $F : \mathcal{RE} \to \mathcal{RE}$ een effectieve operatie. Dan is F continu.

Bewijs. 1. X is r.e. dus $X = W_g$ voor zekere g. We laten zien (weer gebruik makend van opgave 53):

i) $e \in X,\ W_e \subseteq W_f \Rightarrow f \in X$

ii) Er is een primitief recursieve functie S zodat voor alle e:
$e \in X \Rightarrow W_{S(e)}$ eindig, $W_{S(e)} \subseteq W_e$ en $S(e) \in X$

Voor i): neem aan $e \in X$. Kies met de recursiestelling een code h zodat voor alle e, f en x:

$$h \bullet (e, f, x) \simeq \mu z.[T(1, e, x, z) \vee (T(1, f, x, j_1(z)) \wedge T(1, g, S_1^2(h, e, f), j_2(z)))]$$

Nu geldt: als $S_1^2(h, e, f) \notin X$ is $g \bullet S_1^2(h, e, f)$ ongedefinieerd, dus dan is $S_1^2(h, e, f) \bullet x \simeq h \bullet (e, f, x)$ gedefinieerd dan en slechts dan als $e \bullet x$ gedefinieerd is; m.a.w. $W_{S_1^2(h,e,f)} = W_e$ dus $S_1^2(h, e, f) \in X$, tegenspraak; dus $S_1^2(h, e, f) \in X$ en er volgt

$$W_{S_1^2(h,e,f)} = W_e \cup W_f$$

zodat, als $W_e \subseteq W_f$, $W_{S_1^2(h,e,f)} = W_f$ en $f \in X$.
Voor ii): Kies, weer met de recursiestelling, een code f zodat voor alle e, x:

$$f \bullet (e, x) \simeq \begin{cases} e \bullet x & \text{als } \neg\exists z \leq x.T(1, g, S_1^1(f, e), z) \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

Stel $e \in X$. Ga zelf na dat volgt: $S_1^1(f, e) \in X$, $W_{S_1^1(f,e)}$ is eindig, en $W_{S_1^1(f,e)} \subseteq W_e$. We kunnen dus zetten: $S(e) = S_1^1(f, e)$.
2. Dit volgt vrij direkt uit de continuïteitsuitspraak in Myhill-Shepherdson. Definieer functies $H_1 : \mathcal{RE} \to \mathcal{PR}$ en $H_2 : \mathcal{PR} \to \mathcal{RE}$ door:

$$\begin{array}{rcl} H_1(R) & = & \text{de partiële functie met domein } R \text{ en bereik } \{0\} \\ H_2(f) & = & \{x \mid f(x) = 0\} \end{array}$$

Ga zelf na dat er primitief recursieve functies h_1 en h_2 zijn zodat voor alle e geldt $H_1(W_e) = \varphi_{h_1(e)}$ en $H_2(\varphi_e) = W_{h_2(e)}$, en dat de compositie $H_2 \circ H_1$ de identiteit op $\mathcal{RE}$ is. Hieruit volgt, dat als $F : \mathcal{RE} \to \mathcal{RE}$ een effectieve operatie op $\mathcal{RE}$ is, $H_1 \circ F \circ H_2 : \mathcal{PR} \to \mathcal{PR}$ ook een effectieve operatie is, dus continu met de stelling van Myhill-Shepherdson. Nu is

$$F = H_2 H_1 F H_2 H_1 = H_2(H_1 F H_2)H_1$$

dus de continuïteit van F volgt uit de continuïteit van H_1 en H_2, die men gemakkelijk nagaat. ■

Opgave 56. Concludeer uit de stelling van Rice-Shapiro dat er geen totaal recursieve functie F is zodat voor alle e geldt: W_e is een singleton precies dan als $F(e) \in \mathcal{K}$.

Opgave 57. Concludeer uit de stelling van Rice-Shapiro dat $\{x \mid W_x \text{ eindig}\}$ noch $\{x \mid W_x \text{ oneindig}\}$ r.e. is.

Iets lastiger is de situatie voor $\mathcal{R}$. Hier zijn de basis-open verzamelingen van de vorm $\mathcal{O}_u$ zijn, met

$$\mathcal{O}_u = \{f \in \mathcal{R} \mid \forall i < \text{lh}(u)\, f(i) = (u)_i\}$$

Het resultaat over r.e. extensionele verzamelingen ontbreekt in deze situatie. Ten eerste:

Opgave 58. Laat zien: als X een r.e. extensionele verzameling codes van totaal recursieve functies is, is $X = \emptyset$.

Nu kan de vraag iets genuanceerd worden: laten we X *extensioneel met betrekking tot codes van totale functies* noemen, als voor elk tweetal codes e, e' van totale functies geldt: $\varphi_e = \varphi_{e'} \Rightarrow (e \in X \Leftrightarrow e' \in X)$ (X hoeft niet uitsluitend codes van totaal recursieve functies te bevatten). Geldt misschien voor X r.e. en extensioneel m.b.t. codes van totaal recursieve functies, dat $\{\varphi_e \in \mathcal{R} \mid e \in X\}$ open is in $\mathcal{R}$? Het antwoord wordt gegeven in de volgende propositie.

Propositie 2.18 (Friedberg, 1958) *Er bestaat een r.e. verzameling X die extensioneel is m.b.t. codes van totaal recursieve functies, zodat*

$$\{\varphi_e \in \mathcal{R} \mid e \in X\}$$

niet open is in $\mathcal{R}$.

Bewijs. Zij

$$X = \{e \mid (\forall y \leq e.e \bullet y = 0) \vee \exists z[e \bullet z \neq 0 \wedge \forall y < z.e \bullet y = 0 \wedge \exists e' < z \forall u \leq z.e' \bullet u = e \bullet u]\}$$

Hier moet $e \bullet z \neq 0$ gelezen worden als $\exists y[T(1,e,z,y) \wedge U(y) \neq 0]$; ga zelf na dat X r.e. is. Om in te zien dat X extensioneel m.b.t. codes van totaal recursieve functies is, stel $\varphi_e = \varphi_{e'}$ totaal en $e \in X$. De definitie van X is van de vorm (1) $\vee$ (2) waarbij (2) duidelijk een extensionele eigenschap is. Als $\varphi_{e'} = \lambda x.0$ dan $e' \in X$. Anders, laat y' het minimale niet-nulpunt van $\varphi_{e'}$. Nu geldt hetzij $\forall y \leq e.e \bullet y = e' \bullet y = 0$ (oftewel $e \in X$ op grond van (1)) in welk geval $e < y' \leq e'$ (of $e' < y'$ zodat ook $e' \in X$ op grond van (1)) zodat $e' \in X$ op grond van (2), hetzij $e \in X$ op grond van (2) in welk geval $e' \in X$ op grond van (2) wegens (2) extensioneel.
Nu is $T = \{\varphi_e \in \mathcal{R} \mid e \in X\}$ echter niet open in $\mathcal{R}$: wegens $\lambda x.0 \in T$ zou er, ware T open, een u zijn met $\forall i < \mathrm{lh}(u).(u)_i = 0 \wedge \mathcal{O}_u \subseteq T$; kies zo'n u en laat $n = \mathrm{lh}(u)$. Kies een code g zodat voor alle x:

$$g \bullet x \quad \simeq \quad \begin{cases} 0 & \text{als } x \neq n \\ k & \text{anders} \end{cases}$$

waar $k \neq 0$ zo bepaald wordt dat geen partieel recursieve functie met code $< n$ hetzelfde gedrag heeft als φ_g op $\{0,\ldots,n\}$. Er geldt: $\varphi_g \in \mathcal{O}_u$, maar $n \leq g$ (ga na), en $g \notin X$ ∎

Wel geldt het continuïteitsresultaat, waarvoor de belangrijkste bouwsteen de stelling van Kreisel-Lacombe-Shoenfield (onafhankelijk ook bewezen door de Rus Čeitin) is.

Stelling 2.19 (Kreisel, Lacombe, Shoenfield (1957)) *Er is een partieel recursieve functie K met de volgende eigenschap: voor iedere effectieve operatie $F : \mathcal{R} \to \mathbb{N}$ (effectief in de zin dat $F(\varphi_e) = f \bullet e$ voor zekere f) en iedere f met*

$$\{e \mid \varphi_e \text{ totaal}\} \subseteq \mathrm{dom}(\varphi_f) \wedge \forall e(\varphi_e \text{ totaal} \Rightarrow f \bullet e = F(\varphi_e))$$

en voor iedere e met φ_e totaal, is $K(f,e)$ gedefinieerd, en er geldt:

1. $\varphi_e \in \mathcal{O}_{K(f,e)}$
2. $\forall e'(\varphi_{e'} \text{ totaal} \wedge \varphi_{e'} \in \mathcal{O}_{K(f,e)} \Rightarrow F(\varphi_e) = F(\varphi_{e'}))$

Bewijs. Laat $u^\circ = \lambda x.(u)_x$ (Herinner dat $(u)_x = 0$ voor $x \geq \mathrm{lh}(u)$ per afspraak); en G primitief recursief zodat $u^\circ = \varphi_{G(u)}$. We definiëren een partieel recursieve functie van 4 variabelen als volgt:

$$\eta(f,y,z,x) \simeq \begin{cases} \text{ongedefinieerd} & \text{als} & f \bullet y \text{ ongedefinieerd} & (1) \\ y \bullet x & \text{als} & \neg\exists w \leq x(T(1,f,z,w) \wedge U(w) = f \bullet y) & (2) \\ u^\circ(x) & \text{als} & w \leq x \text{ minimaal met } T(1,f,z,w) \wedge U(w) = f \bullet y, & \\ & & \text{en } \langle u,t\rangle \text{ minimaal met } \mathrm{lh}(u) > w \wedge & \\ & & \wedge\ \forall i < w \exists v < t(T(1,y,i,v) \wedge U(v) = (u)_i) \wedge & (3) \\ & & \wedge\ \exists v < t(T(1,f,G(u),v) \wedge U(v) \neq f \bullet y) & \\ \text{ongedefinieerd} & \text{als} & \text{in (3) zo'n } \langle u,t\rangle \text{ niet bestaat} & \end{cases}$$

We kiezen nu met de recursiestelling een z zodat voor alle f, y, x:

$$z \bullet (f,y,x) \simeq \eta(f,y,S^2_1(z,f,y),x)$$

Stel nu dat F een effectieve operatie is, zodat $F(\varphi_e) = f \bullet e$ voor alle codes van totaal recursieve functies e, en stel φ_y totaal. Dan is $f \bullet y$ gedefinieerd. Verder geldt:

i) $\exists w(T(1,f,S^2_1(z,f,y),w) \wedge U(w) = f \bullet y)$. Want stel niet, dan is $S^2_1(z,f,y) \bullet x$ gelijk aan $y \bullet x$ voor alle x (uit de definitie van η, want $S^2_1(z,f,y) \bullet x \simeq \eta(f,y,S^2_1(z,f,y),x)$), m.a.w. $\varphi_{S^2_1(z,f,y)} = \varphi_y$, maar dan moet juist wèl $f \bullet S^2_1(z,f,y) = f \bullet y$ gelden, vanwege de aanname op f.

ii) Zij w minimaal zodat $T(1, f, S^2_1(z, f, y), w)$. Dan bestaat er geen u met

$$w \leq \mathrm{lh}(u) \wedge \forall i < w.y \bullet i = (u)_i \wedge f \bullet G(u) \neq f \bullet y$$

Immers stel wel, dan was er $\langle u, t\rangle$ als in (3) van de definitie van η, en zou volgen $\varphi_{S^2_1(z,f,y)} = \varphi_{G(u)} = u^\circ$ dus $f \bullet S^2_1(z, f, y) = f \bullet G(u) \neq f \bullet y$, in tegenspraak met i).

iii) Met w als in ii): als $\varphi_{y'}$ totaal en $\forall i < w.y' \bullet i = y \bullet i$, dan $f \bullet y = f \bullet y'$; immers als w' zodat $T(1, f, S^2_1(z, f, y'), w')$ en w'' het maximum van w en w':
$f \bullet y' = f \bullet G(\langle y' \bullet 0, \ldots, y' \bullet (w' - 1)\rangle)$ vanwege ii) met y' i.p.v. y; dit is ook gelijk aan $f \bullet G(\langle y' \bullet 0, \ldots, y' \bullet (w'' - 1)\rangle) = f \bullet y$, want $\langle y' \bullet 0, \ldots, y' \bullet (w'' - 1)\rangle$ is een voortzetting van het rijtje $\langle y \bullet 0, \ldots, y \bullet (w - 1)\rangle$.

Dus als we zetten:

$$K(f, y) \simeq \langle y \bullet 0, \ldots, y \bullet (v - 1)\rangle$$

waar $v = \mu w.T(1, f, S^2_1(z, f, y), w)$, dan heeft K de in de stelling geponeerde eigenschappen. Ga na dat K partieel recursief is. ∎

Gevolg 2.20 *Zij $F : \mathcal{R} \to \mathcal{R}$ een effectieve operatie. Dan is F continu.*

Bewijs. We moeten laten zien dat voor elke $f \in \mathcal{R}$ en elke u met $F(f) \in \mathcal{O}_u$, er een v is zodat $f \in \mathcal{O}_v \wedge F[\mathcal{O}_v] \subseteq \mathcal{O}_u$.
Omdat F een effectieve operatie is, is er een partieel recursieve functie ψ zodat voor alle i en alle codes van totaal recursieve functies e:

$$\psi(e, i) = F(\varphi_e)(i)$$

Laat G primitief recursief zodat $G(i)$ een code van $\lambda e.\psi(e, i)$ is; er geldt:

$$\varphi_e = \varphi_{e'} \Rightarrow G(i) \bullet e = G(i) \bullet e'$$

voor alle i en alle codes van totaal recursieve functies e, e'; dus we kunnen stelling 2.19 op $G(i)$ toepassen. Zij $f \in \mathcal{R}$, $f = \varphi_y$. Laat

$$w = \max\{\mathrm{lh}(K(G(i), y)) \mid 0 \leq i < \mathrm{lh}(u)\}$$

Dan geldt

$$g \in \mathcal{O}_{\langle y \bullet 0, \ldots, y \bullet (w-1)\rangle} \Rightarrow F(g) \in \mathcal{O}_u$$

(ga na) ∎

Opmerking Het getal $w = w_{y,u} = \max\{\mathrm{lh}(K(G(i), y)) \mid 0 \leq i < \mathrm{lh}(u)\}$ wordt wel een *modulus van continuïteit* van F bij φ_y genoemd. Het hangt echter in de eerste plaats van de *code* y af; d.w.z. er is in het algemeen geen partieel recursieve functie N die voldoet aan: $N(y, u)$ is een modulus van continuïteit, en $\varphi_y = \varphi_{y'} \Rightarrow N(y, u) = N(y', u)$.

Opgave 59. Laat zien dat er geen effectieve operatie $F : \mathcal{R} \to \mathbb{N}$ is zodat voor alle codes van totaal recursieve functies e geldt:

$$0 \in \mathrm{rge}(\varphi_e) \Leftrightarrow F(e) \in \mathcal{K}$$

2.4 Stricte r.e. verzamelingen in de wiskunde en de logica

In deze paragraaf bespreek ik een paar voorbeelden van problemen in wiskunde en logica, waarvan de onoplosbaarheid middels recursietheorie kon worden aangetoond. Steeds is er sprake van een geschikte *codering* van het probleem in termen van natuurlijke getallen.

2.4.1 Hilberts tiende probleem

In 1900 presenteerde David Hilbert een lijst met problemen die de wiskunde volgens hem in de 20e eeuw diende op te lossen. Het tiende probleem handelde over *Diophantische vergelijkingen*, dat zijn vergelijkingen van de vorm $P(X_1, \ldots, X_n) = 0$, waar $P \in \mathbb{Q}[X_1, \ldots, X_n]$ een polynoom met rationale coëfficiënten is. De vraag luidde:

> 10. Gegeven een Diophantische vergelijking in een willekeurig aantal variabelen en met gehele coëfficiënten; ontwerp een procedure met behulp waarvan in een eindig aantal stappen uitgemaakt kan worden of de vergelijking een gehele oplossing heeft.

Nu kan elk polynoom in n variabelen natuurlijk gecodeerd worden als rijtje $n+1$-tallen (voor coëfficiënten en machten), bijvoorbeeld: codeer $4X_1^4X_2 + 5X_2^2X_3 + X_1X_3$ als

$$\langle\langle\ulcorner 4\urcorner, 4, 1, 0\rangle, \langle\ulcorner 5\urcorner, 0, 2, 1\rangle, \langle\ulcorner 1\urcorner, 1, 0, 1\rangle\rangle$$

(waar $\ulcorner - \urcorner$ een geschikte codering: $\mathbb{Q} \to \mathbb{N}$ geeft) zodat (als $\ulcorner P\urcorner$ de code van polynoom P is) de functie

$$\lambda x_1 \cdots x_n.P(x_1, \ldots, x_n)$$

primitief recursief in $\ulcorner P\urcorner$ en $x_1, \ldots, x_n$ is. De verzameling

$$\{u \mid u = \ulcorner P\urcorner \wedge \exists\vec{x}.P(\vec{x}) = 0\}$$

is dan duidelijk r.e. Hilberts vraag veronderstelt dat de verzameling van alle u zodat u een polynoom P in n variabelen codeert waarvoor $\exists x_1 \cdots x_n P(x_1, \ldots, x_n) = 0$, recursief is.

In 1970 bewees Matijasevič dat er voor elke e en n een polynoom P in minstens $n+1$ variabelen bestaat, zodat

$$W_e^n = \{(x_1, \ldots, x_n) \mid \exists a_{n+1} \cdots a_{n+k} P(x_1, \ldots, x_n, a_{n+1}, \ldots, a_{n+k}) = 0\}$$

Er geldt dan

$$W_e^n \neq \emptyset \Longleftrightarrow \exists x_1 \cdots x_{n+k} P(x_1, \ldots, x_{n+k}) = 0$$

Een oplossing van Hilberts tiende probleem in zijn volle algemeenheid zou ons dus een beslissingsprocedure geven voor de vraag: is $W_e^n = \emptyset$? Maar we hebben gezien dat zo'n procedure niet kan bestaan.

Wie geïnteresseerd is in het bewijs van Matijasevič' stelling, kan terecht bij het boek van Smoryński.

2.4.2 Woord-problemen: groepen

Woord-problemen treden op bij elke mathematische structuur die gedefinieerd kan worden in termen van voortbrengers en relaties. Als voorbeeld schets ik het woord-probleem voor groepen.

Zij $A = \{a_1, \ldots, a_n\}$ een eindige verzameling; associeer met elke a_i een nieuw element $a_i^{-1} \in A^{-1} = \{a_1^{-1}, \ldots, a_n^{-1}\}$; zodat $a_i^{-1} \neq a_j^{-1}$ als $a_i \neq a_j$.

Een *woord* op A is een rijtje

$$b_1 \cdots b_m \tag{1}$$

waarvoor geldt dat alle $b_i \in A \cup A^{-1}$ voor $1 \leq i \leq m$, en waarin nooit a_i en a_i^{-1} naast elkaar staan. De verzameling woorden op A schrijven we als $[A]$. In (1) laten we $m = 0$ toe; we spreken van het lege woord.

Er is een simpele operatie op woorden: concatenatie. Als $\sigma = b_1 \cdots b_m$ en $\tau = c_1 \cdots c_l$ laat $\sigma * \tau = b_1 \cdots b_m c_1 \cdots c_l$ zijn; dit hoeft geen woord te zijn, maar door eindig vaak een aanliggend paar $a_i a_i^{-1}$ of $a_i^{-1} a_i$ uit $\sigma * \tau$ te verwijderen ontstaat wèl een woord, dat we noteren met $\sigma\tau$.

Propositie 2.21 *Met de operatie $\sigma, \tau \mapsto \sigma\tau$ is $[A]$ een groep.*

Bewijs. Zie, bijvoorbeeld, het boek *Combinatorial Group Theory* van R.C. Lyndon en P.E.Schupp, Springer 1977 (herdrukt in 2001). ■

$[A]$ heet de *vrije* groep voortgebracht door de verzameling A.

Een *relatie* op A is een uitdrukking $X = Y$, waar $X, Y \in [A]$. Als R een verzameling relaties op A is, geven we met $[A; R]$ de factorgroep $[A]/K$ aan, waar K de normaaldeler van $[A]$ is die voortgebracht wordt door $\{X^{-1}Y \mid (X = Y) \in R\}$.

Een *eindige presentatie* van een groep G is een paar (A, R) met A een eindige verzameling en R een eindige verzameling relaties op A, zodat $G \cong [A; R]$. We zeggen: G is eindig gepresenteerd.

Laat $\phi_R : [A] \to [A; R]$ de quotiëntafbeelding zijn. We zeggen dat het *woord-probleem voor* $G = [A; R]$ *oplosbaar is*, als er een procedure is die voor elk tweetal woorden X, Y op A uitmaakt of $\phi_R(X) = \phi_R(Y)$ in $[A; R]$.

Zo op het eerste gezicht lijkt dit af te hangen van de presentatie; maar men kan laten zien dat het al of niet oplosbaar zijn van het woord-probleem niet van de presentatie afhangt.

De *stelling van Novikov* (1955) luidt, dat er een eindig gepresenteerde groep is met een onoplosbaar woord-probleem. Wie het bewijs wil lezen, zij verwezen naar de Appendix van:
J.R. Shoenfield, *Mathematical Logic*, Addison-Wesley 1967 (of de recente reprint: Association for Symbolic Logic, Massachusetts 2001). Ook het eerder aangehaalde boek *Combinatorial Group Theory* van Lyndon en Schupp bevat hier materiaal over.

2.4.3 De stellingen van Church en Trachtenbrot

In hoofdstuk 5 van deze syllabus zullen we nog uitvoerig zien hoe formules en bewijzen in de predicatenlogica gecodeerd kunnen worden door natuurlijke getallen. Als dit "eenvoudig" genoeg gebeurt, is de verzameling codes van bewijsbare formules duidelijk r.e.

De *Stelling van Church*, die we zullen bewijzen, zegt dat deze verzameling strict r.e. is: anders gezegd: de predicatenlogica is *onbeslisbaar*, er is geen procedure die voor een gegeven formule φ uitmaakt of φ bewijsbaar is in de predicatenlogica.

De stelling van Trachtenbrot geeft een voorbeeld van recursief onscheidbare, disjuncte r.e. verzamelingen buiten de recursietheorie: evenals formules kunnen ook eindige structuren in getallen worden gecodeerd, en wel zodat de verzameling A van paren (x, y) zodat x een eindige structuur $\aleph$ codeert en y een zin φ, en $\aleph \not\models \varphi$, recursief is.

Zij A_1 de verzameling codes van bewijsbare zinnen uit de predicatenlogica; de stelling van Trachtenbrot beweert, dat A_1 en

$$A_2 = \{y \mid \exists x((x, y) \in A)\}$$

recursief onscheidbare r.e. verzamelingen zijn. Voor het bewijs: zie Smoryński.

3 Reductie en Klassificatie

Dit hoofdstuk gaat over *definieerbare* deelverzamelingen van $\mathbb{N}^k$. Een deelverzameling $A \subseteq \mathbb{N}^k$ heet definieerbaar, als de uitspraak $\vec{x} \in A$ equivalent is aan een uitspraak die kan worden opgeschreven met behulp van alleen primitief recursieve predicaten, logische tekens als $\neg, \wedge, \vee, \Rightarrow$ en kwantoren $\exists, \forall$. Elke r.e. verzameling $A \subseteq \mathbb{N}^k$ is definieerbaar, want er is een primitief recursief predicaat $S \subseteq \mathbb{N}^{k+1}$ zodat $\vec{x} \in A$ equivalent is met $\exists y.(\vec{x}, y) \in S$.

De definieerbare deelverzamelingen kunnen worden geklassificeerd naar de complexiteit van de eenvoudigst mogelijke definitie; we krijgen een gelaagde structuur die *arithmetische hiërarchie* heet.

Een belangrijk instrument om de precieze plaats van een verzameling binnen de arithmetische hiërarchie te bepalen, is het begrip 'reduceerbaar' dat we kennen van hoofdstuk 2, en dat we hier nader beschouwen. Voorts blijkt het mogelijk, om met behulp van deze reduceerbaarheid de structuur van de arithmetische hiërarchie te verfijnen; iets hiervan werken we uit voor r.e. verzamelingen (zie 3.3).

3.1 Many-one reduceerbaarheid

Definitie 3.1 Een *preordening* op een verzameling is een reflexieve en transitieve relatie.

Een preordeningsrelatie wordt vaak aangegeven met $\leq$ alsof het een ordening is. Ga na, dat als $\leq$ een preordening op X aangeeft, de relatie $\equiv$, gedefinieerd door:

$$x \equiv y \text{ precies dan als } x \leq y \wedge y \leq x$$

een equivalentierelatie op X is. De preordening $\leq$ induceert dan een partiële ordening op de verzameling equivalentieklassen modulo $\equiv$.

Definitie 3.2 Laat $\leq$ een preordening zijn op de verzameling X, en $x, y, z \in X$. We zeggen dat z een *supremum* (of kleinste bovengrens) van x en y is, als voor elke $w \in X$ geldt:

$$z \leq w \Longleftrightarrow x \leq w \wedge y \leq w$$

Merk op, dat in een preordening een supremum van x en y hooguit op $\equiv$-equivalentie na vastligt.

Een verzameling X met een preordening $\leq$ waarvoor geldt, dat elk tweetal elementen van X een supremum heeft, noemen we een *pre-halftralie*.

We definiëren nu de volgende preordening op $\mathcal{P}(\mathbb{N})$:

Definitie 3.3 Stel X en Y zijn deelverzamelingen van $\mathbb{N}$. We noemen X *many-one reduceerbaar* tot Y, notatie $X \leq_m Y$, als er een totaal recursieve functie F is zodat

$$X = F^{-1}(Y) = \{x \mid F(x) \in Y\}$$

Gegeven zo'n F, heet X ook wel many-one reduceerbaar tot Y *via* F.

We schrijven $X \equiv_m Y$ voor de bijbehorende equivalentierelatie: $X \equiv_m Y$ als $X \leq_m Y$ en $Y \leq_m X$.

Het subscript m slaat op "many-one", en beklemtoont dat de gevraagde functie F niet injectief hoeft te zijn.

In dit hoofdstuk bestuderen we de relatie $\leq_m$. Ik vat de volgende elementaire eigenschappen van $\leq_m$ samen in een opgave:

Opgave 60.

a) Bewijs, dat $\leq_m$ een preordening op $\mathcal{P}(\mathbb{N})$ is;

b) Bewijs: $X \leq_m Y \Leftrightarrow X^c \leq_m Y^c$;

c) Bewijs, dat voor alle $X \subseteq \mathbb{N}$ geldt:

$$\begin{array}{lcl} \mathbb{N} \leq_m X & \Leftrightarrow & X \neq \emptyset \\ \emptyset \leq_m X & \Leftrightarrow & X \neq \mathbb{N} \end{array}$$

d) Stel $X \neq \emptyset$, $X \neq \mathbb{N}$ en X is recursief. Bewijs, dat X een supremum is van $\emptyset$ en $\mathbb{N}$;

e) Stel $X, Y \subseteq \mathbb{N}$. Definieer de verzameling $X \sqcup Y$ door:

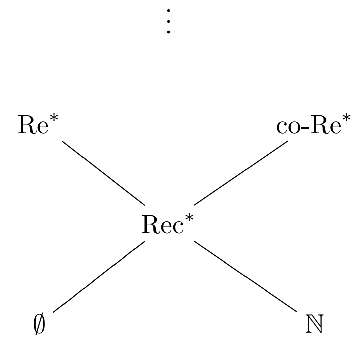
$$X \sqcup Y \ = \{2x \,|\, x \in X\} \cup \{2x+1 \,|\, x \in Y\}$$

Bewijs, dat $X \sqcup Y$ een supremum is van X en Y;

f) Stel $X \leq_m Y$. Bewijs: als Y recursief is, is X het ook; en als Y r.e. is, is X het ook;

g) Bewijs: als X r.e. is en X is niet recursief, dan $X \not\leq_m X^c$.

Uit deze opgave blijkt onder meer, dat $\mathcal{P}(\mathbb{N})$ met de relatie $\leq_m$ een pre-halftralie vormt. De "onderkant" van de preordening $\leq_m$ ziet er zo uit:



waar:

$$\begin{array}{rcl} \mathrm{Rec}^* & = & \{X \,|\, X \neq \emptyset, X \neq \mathbb{N}, X \text{ recursief}\} \\ \mathrm{Re}^* & = & \{X \,|\, X \text{ r.e., niet recursief}\} \\ \text{co-Re}^* & = & \{X \,|\, X^c \text{ r.e., niet recursief}\} \end{array}$$

Verder is Rec^* een *cluster* m.b.t. $\leq_m$: dat wil zeggen dat voor alle $X, Y \in \mathrm{Rec}^*$ geldt, dat $X \equiv_m Y$. Later zullen we zien, dat dit *niet* waar is voor Re^*!

De volgende opgave is om wat te oefenen met de relatie $\leq_m$.

Opgave 61. Bewijs:

$$\{x \,|\, W_x \text{ is oneindig}\} \equiv_m \{x \,|\, \varphi_x \text{ is totaal}\} \equiv_m \{x \,|\, \mathrm{rge}(\varphi_x) \text{ is oneindig}\}$$

Stel nu, dat $\mathcal{X}$ een collectie deelverzamelingen van $\mathbb{N}$ is. $X \in \mathcal{X}$ heet *m-compleet* in $\mathcal{X}$, als voor alle $Y \in \mathcal{X}$ geldt, dat $Y \leq_m X$ (X is "het grootste element" van $\mathcal{X}$).

Opgave 62. Bewijs, dat $\mathcal{K}$ m-compleet is in Re^*.

Via rijtjes-codering breiden we de relatie $\leq_m$ uit tot verzamelingen van k- en n-tallen: voor $X \subseteq \mathbb{N}^k$ en $Y \subseteq \mathbb{N}^n$ zeggen we $X \leq_m Y$ als

$$\{j^k(\vec{x}) \,|\, \vec{x} \in X\} \leq_m \{j^n(\vec{y}) \,|\, \vec{y} \in Y\}$$

Ga na dat dit equivalent is met: er zijn totaal recursieve functies $F_1, \ldots, F_n : \mathbb{N}^k \to \mathbb{N}$, zodat

$$X = \{\vec{x} \,|\, (F_1(\vec{x}), \ldots, F_n(\vec{x})) \in Y\}$$

3.2 De arithmetische hiërarchie

Definitie 3.4 De klassen van Σ_n-, Π_n- en Δ_n-relaties $R \subseteq \mathbb{N}^k$ (voor variabele k) zijn gedefinieerd als volgt:

1. $R \subseteq \mathbb{N}^k$ is een Σ_0-relatie $\Leftrightarrow$ R is een Π_0-relatie $\Leftrightarrow$ R is primitief recursief;
2. R is een Σ_{n+1}-relatie dan en slechts dan als er een Π_n-relatie R' is zodat
$$R = \{\vec{x} \mid \exists y((\vec{x}, y) \in R')\}$$
3. R is een Π_{n+1}-relatie dan en slechts dan als er een Σ_n-relatie R' is zodat
$$R = \{\vec{x} \mid \forall y((\vec{x}, y) \in R')\}$$
4. R is een Δ_n-relatie dan en slechts dan als R zowel een Σ_n- als een Π_n-relatie is.

We schrijven ook wel Σ_n (Π_n, Δ_n) voor de klasse van Σ_n (c.q. Π_n-,Δ_n-)-relaties; en de schrijfwijzen "R is een Σ_n-relatie" en "$R \in \Sigma_n$" zijn equivalent.

Merk op:

- Δ_1 is de klasse van recursieve relaties;
- Σ_1 is de klasse van r.e. relaties;
- Π_1 is de klasse van complementen van r.e. relaties

Propositie 3.5 $\Sigma_n \cup \Pi_n \subseteq \Delta_{n+1}$

Bewijs. Dit berust op het logische trucje van het invoeren van "dummy-variabelen": voor $S \subseteq \mathbb{N}^m$, laat $S \times \mathbb{N} = \{(x_1, \ldots, x_{m+1}) \mid (x_1, \ldots, x_m) \in S\}$. Dan is
$$S = \{\vec{x} \mid \forall x_{m+1}((\vec{x}, x_{m+1}) \in S \times \mathbb{N})\} = \{\vec{x} \mid \exists x_{m+1}((\vec{x}, x_{m+1}) \in S \times \mathbb{N})\}$$
Met inductie is makkelijk na te gaan: $S \in \Sigma_n(\Pi_n) \Rightarrow S \times \mathbb{N} \in \Sigma_n(\Pi_n)$. De propositie volgt nu met inductie:
$\Sigma_0 = \Sigma_0 \cup \Pi_0 \subseteq \Delta_1$ want alle primitief recursieve relaties zijn recursief;
Stel $S \in \Sigma_{n+1}$, dus $S = \{\vec{x} \mid \exists y((\vec{x}, y) \in S')\}$ voor zekere $S' \in \Pi_n$. Dan
$$S = \{\vec{x} \mid \forall z \exists y((\vec{x}, y, z) \in S' \times \mathbb{N})\} \in \Pi_{n+2}$$
en $S \in \Sigma_{n+2}$ is duidelijk, want $S' \in \Pi_{n+1}$ per inductiehypothese.
Het argument voor $S \in \Pi_{n+1}$ is identiek. ∎

We krijgen het volgende plaatje:

$$\begin{tikzcd}
 & & \Sigma_1 \arrow[dr, dash, "\subseteq"] & & \Sigma_2 \arrow[dr, dash, "\subseteq"] & & {} & \\
\Sigma_0 = \Pi_0 = \Delta_0 \arrow[r, dash, "\subseteq"] & \Delta_1 \arrow[ur, dash, "\subseteq"] \arrow[dr, dash, "\subseteq"'] & & \Delta_2 \arrow[ur, dash, "\subseteq"] \arrow[dr, dash, "\subseteq"'] & & \Delta_3 \arrow[ur, dash] \arrow[dr, dash] & & \cdots \\
 & & \Pi_1 \arrow[ur, dash, "\subseteq"'] & & \Pi_2 \arrow[ur, dash, "\subseteq"'] & & {} &
\end{tikzcd}$$

Ons voornaamste doel is nu, aan te tonen dat alle inclusies in dit plaatje *strict* zijn (d.w.z. in feite hebben we overal $\subsetneq$); dit is de zg. hiërarchie-stelling (3.11); in het bijzonder volgt hieruit, dat $\Sigma_n \neq \Pi_n$ (waarom?).

Voorbeelden.
$$\{x \mid \varphi_x \text{ is totaal}\} = \{x \mid \forall y \exists z T(1, x, y, z)\} \in \Pi_2$$
$$\{x \mid W_x \text{ is eindig}\} = \{x \mid \exists n \forall z(T(1, x, j_1(z), j_2(z)) \Rightarrow j_1(z) \leq n)\} \in \Sigma_2$$

Propositie 3.6 *Stel $n > 0$. Als $R \in \Sigma_n$ en $S \leq_m R$ dan $S \in \Sigma_n$.*

Opgave 63.

a) Bewijs propositie 3.6. Gebruik inductie.

b) Bewijs, dat propositie 3.6 ook geldt met Σ_n vervangen door Π_n.

c) Waarom is de voorwaarde $n > 0$ in propositie 3.6 nodig?

Propositie 3.7

i) Als $R, S \in \Sigma_n(\Pi_n)$ dan $R \cap S, R \cup S \in \Sigma_n(\Pi_n)$;

ii) Als $R \subseteq \mathbb{N}^{k+1}$ een $\Sigma_n(\Pi_n)$-relatie is, dan zijn

$$\{(\vec{x}, y) \mid \exists w \leq y(\vec{x}, w) \in R\} \text{ en}$$

$$\{(\vec{x}, y) \mid \forall w \leq y(\vec{x}, w) \in R\}$$

het ook;

iii) $(n > 0)$ Als $R \subseteq \mathbb{N}^{k+1}$ een Σ_n-relatie is, dan is $\{\vec{x} \mid \exists w(\vec{x}, w) \in R\}$ het ook;
Als $R \subseteq \mathbb{N}^{k+1}$ een Π_n-relatie dan ook $\{\vec{x} \mid \forall w(\vec{x}, w) \in R\}$.

Bewijs.

i) Voor $n = 0$ is dit duidelijk. Stel nu $R, S \in \Sigma_{n+1}$; dan zijn er $R', S' \in \Pi_n$ zodat

$$R = \{\vec{x} \mid \exists y(\vec{x}, y) \in R'\},\ S = \{\vec{x} \mid \exists w(\vec{x}, w) \in S'\}$$

dus

$$R \cap S = \{\vec{x} \mid \exists y \exists w((\vec{x}, y) \in R' \wedge (\vec{x}, w) \in S')\}$$

Nu is $\{(\vec{x}, y, w) \mid (\vec{x}, y) \in R' \wedge (\vec{x}, w) \in S'\} \in \Pi_n$ per inductiehypothese, dus met propositie 3.6 volgt

$$\{(\vec{x}, z) \mid (\vec{x}, j_1(z)) \in R' \wedge (\vec{x}, j_2(z)) \in S'\} \in \Pi_n$$

zodat

$$R \cap S = \{\vec{x} \mid \exists z((\vec{x}, j_1(z)) \in R' \wedge (\vec{x}, j_2(z)) \in S')\} \in \Sigma_{n+1}$$

En $R \cup S = \{\vec{x} \mid \exists y((\vec{x}, y) \in R' \cup S')\} \in \Sigma_{n+1}$, want $R' \cup S' \in \Pi_n$ per inductiehypothese. Voor $R, S \in \Pi_{n+1}$: pas De Morgan-dualiteit toe. Immers $R \in \Sigma_n \Leftrightarrow R^c \in \Pi_n$; en $R \cap S = (R^c \cup S^c)^c$.

ii) Stel $R \in \Sigma_{n+1}$. Dan geldt voor zekere $R' \in \Pi_n$, dat $R = \{\vec{x} \mid \exists y(\vec{x}, y) \in R'\}$. Er volgt

$$\begin{aligned}
&\{(\vec{x}, y) \mid \forall w \leq y(\vec{x}, w) \in R\} = \\
&\{(\vec{x}, y) \mid \forall w \leq y \exists z(\vec{x}, w, z) \in R'\} = \\
&\{(\vec{x}, y) \mid \exists \sigma(\mathrm{lh}(\sigma) = y + 1 \wedge \forall i \leq y(\vec{x}, i, (\sigma)_i) \in R')\}
\end{aligned}$$

Pas nu inductiehypothese en propositie 3.6 toe.

iii) Gebruik paar-codering en propositie 3.6

■

Om te bepalen of een relatie R een Σ_n- of Π_n-relatie is, past men eerst het zg. *Tarski-Kuratowski algoritme* toe: schrijf R als logische combinatie (met $\wedge, \vee, \neg, \rightarrow, \exists, \forall$) van primitief recursieve relaties. Breng dan de kwantoren naar voren met de van de logica bekende regeltjes:

i) $\neg \forall x \varphi \Leftrightarrow \exists x \neg \varphi$
$\neg \exists x \varphi \Leftrightarrow \forall x \neg \varphi$

ii) $(\forall x\varphi) \wedge \psi \Leftrightarrow \forall x(\varphi \wedge \psi)$ en $(\forall x\varphi) \vee \psi \Leftrightarrow \forall x(\varphi \vee \psi)$ als x niet in ψ voorkomt; dito voor $\exists$

iii) $((\forall x\varphi) \to \psi) \Leftrightarrow \exists x(\varphi \to \psi)$ en
$((\exists x\varphi) \to \psi) \Leftrightarrow \forall x(\varphi \to \psi)$, als x niet in ψ voorkomt.

De voorwaarden in ii) en iii) betekenen dat men soms variabelen moet "ombenoemen" (d.w.z. $\forall y\varphi(y)$ schrijven in plaats van $\forall x\varphi(x)$). De formule wordt in prenex-normaalvorm gebracht, d.w.z. alle kwantoren staan vooraan. En nu:

- begint de formule met $\exists$ en zijn er n kwantorwisselingen, is de bijbehorende relatie Σ_{n+1};
- begint de formule met $\forall$ en zijn er n kwantorwisselingen, is de bijbehorende relatie Π_{n+1};
- is de formule kwantorvrij, is de bijbehorende relatie Σ_0.

Voorbeeld. $\{x \mid \varphi_x$ is totaal en begrensd$\} =$

$$\begin{aligned}
&= \{x \mid \forall y \exists z T(1,x,y,z) \wedge \exists n \forall y \forall z (T(1,x,y,z) \to U(z) \le n)\} \\
&= \text{(variabelen ombenoemen)} \\
& \{x \mid \forall y \exists z T(1,x,y,z) \wedge \exists n \forall v \forall w (T(1,x,v,w) \to U(w) \le n)\} \\
&= \{x \mid \forall y \exists z (T(1,x,y,z) \wedge \exists n \forall v \forall w (\cdots))\} \\
&= \{x \mid \forall y \exists z \exists n \forall v \forall w (\cdots)\}
\end{aligned}$$

met $(\cdots)$ recursief; dus dit is een Π_3-relatie. Maar na het ombenoemen van de variabelen kunnen we ook eerst de kwantoren van het tweede conjunct naar voren halen. We krijgen dan:

$$\{x \mid \exists n \forall v \forall w \forall y \exists z (\cdots)\}$$

en dit is een Σ_3-relatie.

Het Tarski-Kuratowski algoritme levert ons dus altijd een Σ_n of Π_n waar onze relatie in zit (als we haar logisch kunnen definiëren), maar niet altijd de "zuinigste", en vertelt ons niets over Δ_n-relaties zoals deze in het voorbeeld (blijkens de twee berekeningen is dit een Δ_3-relatie).

Om te bewijzen dat een bepaalde relatie R (waarvan men bijvoorbeeld weet dat het een Δ_{n+1}-relatie is) *niet* Σ_n of Π_n is, biedt het algoritme geen soelaas, en moeten we aan het werk:

Voorbeeld. De relatie $\{x \mid \varphi_x$ is totaal en begrensd$\}$ is geen Π_2-relatie.

Bewijs. Stel wel, dan geldt voor zekere recursieve relatie $R(x,y,z)$:

$$\varphi_x \text{ is totaal en begrensd} \Leftrightarrow \forall y \exists z R(x,y,z)$$

Kies nu met de recursiestelling een code e zodat voor alle y:

$$e \bullet y \simeq \max(\{w \le y \mid \forall v \le w \exists z \le y R(e,v,z)\})$$

met de conventie dat $\max(\emptyset) = 0$. Dan is φ_e totaal. Echter, uit de keuze van e kun je nagaan dat

$$\varphi_e \text{ is begrensd} \Leftrightarrow \exists y \forall z \neg R(e,y,z)$$

Aan de andere kant volgt uit de keuze van R:

$$\exists y \forall z \neg R(e,y,z) \Leftrightarrow \varphi_e \text{ is niet totaal en begrensd}$$

zodat we een tegenspraak hebben. ∎

Opgave 64. Bewijs dat $\{x \mid \varphi_x$ is totaal en begrensd$\}$ ook niet Σ_2 is.
(Hint: de definitie van "totaal en begrensd" bestaat uit een Σ_2-deel en een Π_2-deel. In het voorbeeld wordt (om te bewijzen dat de relatie niet Π_2 is, gewoekerd met de Σ_2-component. Hier moet je juist de andere component gebruiken)

Het laatste voorbeeld toont in ieder geval aan, dat de inclusie $\Pi_2 \subseteq \Delta_3$ strict is. We zullen nu inzien dat *elke* inclusie: $\Pi_n \subseteq \Delta_{n+1}$, $\Sigma_n \subseteq \Delta_{n+1}$, $\Delta_n \subseteq \Sigma_n$, $\Delta_n \subseteq \Pi_n$ strict is.

Definitie 3.8 Definieer voor elke $n, k \geq 1$ een Σ_n-relatie $E_n^{(k)} \subseteq \mathbb{N}^{k+1}$ als volgt:

$$E_n^{(k)} = \{(z, x_1, \ldots, x_k) \mid \exists y_1 \forall y_2 \cdots \exists y_n T(n+k-1, z, j^{n+k-1}(x_1, \ldots, x_k, y_1, \ldots, y_{n-1}), y_n)\}$$

als n oneven is, en

$$E_n^{(k)} = \{(z, x_1, \ldots, x_k) \mid \exists y_1 \forall y_2 \cdots \forall y_n \neg T(n+k-1, z, j^{n+k-1}(x_1, \ldots, x_k, y_1, \ldots, y_{n-1}), y_n)\}$$

als n even is.

Stelling 3.9 (Kleene normaalvorm-stelling) *($n > 0$) Elke Σ_n-relatie is van de vorm*

$$\{\vec{x} \mid (z, \vec{x}) \in E_n^{(k)}\}$$

voor zekere z, en elke Π_n-relatie is van de vorm

$$\{\vec{x} \mid (z, \vec{x}) \notin E_n^{(k)}\}$$

voor zekere z.

Als $R \subseteq \mathbb{N}^k$ een Σ_n-relatie is, heet een voorstelling van R als $\{\vec{x} \mid (z, \vec{x}) \in E_n^{(k)}\}$ een Σ_n-*normaalvorm* van R, en z een Σ_n-*code* van R. Analoog hebben we ook Π_n-normaalvormen en -codes.

Bewijs. De bewering over de Π_n-relaties volgt natuurlijk uit die over de Σ_n-relaties. We bewijzen die met inductie naar n:

Elke Σ_1-relatie is $W_e^{(k)}$ voor zekere e oftewel $\{\vec{x} \mid \exists y T(k, e, j^k(\vec{x}), y)\}$.

Stel R is een Σ_{n+1}-relatie, dan is $R = \{\vec{x} \mid \exists y S(\vec{x}, y)\}$ voor een zekere Π_n-relatie S; dus S^c is Σ_n en met inductiehypothese geldt dus $(\vec{x}, y) \notin S \Leftrightarrow (z, \vec{x}, y) \in E_n^{(k+1)}$ voor zekere z. Neem nu aan dat n oneven is (het andere geval gaat net zo); we krijgen:

$$\begin{array}{ll} \vec{x} \in R & \Leftrightarrow \\ \exists y (z, \vec{x}, y) \notin E_n^{(k+1)} & \Leftrightarrow \\ \exists y \neg \exists y_1 \forall y_2 \cdots \exists y_n T(k+n, z, j^{k+n}(\vec{x}, y, y_1, \ldots, y_{n-1}), y_n) & \Leftrightarrow \\ \exists y \forall y_1 \cdots \forall y_n \neg T(k+n, z, j^{k+n}(\vec{x}, y, y_1, \ldots, y_{n-1}), y_n) & \Leftrightarrow \\ (z, \vec{x}) \in E_{n+1}^{(k)} & \end{array}$$

■

Propositie 3.10 *$E_n^{(k)}$ is Σ_n maar niet Π_n.*

Bewijs. (neem aan: $k = 1$) Stel $E_n^{(1)} \in \Pi_n$, dan is ook $\{x \mid (x, x) \in E_n^{(1)}\} \in \Pi_n$, en dus is er een z (met de normaalvorm-stelling) zodat voor alle x:

$$(x, x) \in E_n^{(1)} \Leftrightarrow (z, x) \notin E_n^{(1)}$$

Hieruit volgt natuurlijk voor $x = z$ een contradictie. ■

Gevolg 3.11 (Hiërarchie-stelling) *Alle inclusies in de arithmetische hiërarchie zijn strict.*

Bewijs. Immers, $E_n^{(k)} \in \Sigma_n \setminus \Pi_n$ impliceert $\Delta_n \subsetneqq \Sigma_n$ en $\Pi_n \subsetneqq \Delta_{n+1}$; en uit $(E_n^{(k)})^c \in \Pi_n \setminus \Sigma_n$ volgt $\Delta_n \subsetneqq \Pi_n$ en $\Sigma_n \subsetneqq \Delta_{n+1}$. ■

Opgave 65. Bewijs: als R m-compleet is in Σ_n, dan $R \notin \Pi_n$.

Opgave 66. Bewijs: als A m-compleet is in Σ_n en B m-compleet in Π_n, dan geldt $A \sqcup B \in \Delta_{n+1} \setminus \Sigma_n \cup \Pi_n$.

Propositie 3.12 *De relaties $E_n^{(k)}$ zijn m-compleet in Σ_n en hun complementen zijn dus m-compleet in Π_n.*

Bewijs. Dit volgt onmiddellijk uit de normaalvormstelling (stelling 3.9); ga na. ∎

Het *klassificeren* van een verzameling $X \subseteq \mathbb{N}^k$ is het bepalen van zijn exacte plaats in de arithmetische hiërarchie. Vaak gebeurt dit als volgt: via het Tarski-Kuratowski algoritme probeert men een zo klein mogelijke n te vinden zodat $X \in \Sigma_n$, Π_n of Δ_n; weet men dat $X \in \Sigma_n$ en lukt het niet, te laten zien dat $X \in \Pi_n$, dan probeert men te laten zien dat voor zekere k, $E_n^{(k)} \leq_m X$.

Voorbeeld. $\{x \mid \varphi_x \text{ is totaal}\}$ is strict Π_2.
Bewijs. Noem deze verzameling X. We laten zien dat $(E_2^{(1)})^c \leq_m X$:
$(E_2^{(1)})^c = \{(z,x) \mid \forall y_1 \exists y_2 T(2,z,j(x,y_1),y_2)\}$. Als g een code is zodat voor alle z, x, y_1:

$$g \bullet (z,x,y_1) \simeq z \bullet (x,y_1)$$

dan geldt:

$$\begin{array}{ll} (w_1,w_2) \notin E_2^{(1)} & \Leftrightarrow \\ \forall y_1 \exists y_2 T(2,w_1,j(w_2,y_1),y_2) & \Leftrightarrow \\ \forall y_1 \exists y_2 T(1,S_1^2(g,w_1,w_2),y_1,y_2) & \Leftrightarrow \\ \varphi_{S_1^2(g,w_1,w_2)} \text{ is totaal} & \end{array}$$

Dus $(E_2^{(1)})^c \leq_m X$, als verlangd. ∎

Opgave 67. Laat zien, dat $\{e \mid W_e \text{ is een singleton}\}$ strict Δ_2 is.

Opgave 68. Vind zo klein mogelijke n zodat de volgende relaties in Σ_n, Π_n of Δ_n zijn:

i) $\{e \mid W_e \text{ is eindig}\}$

ii) $\{e \mid \mathrm{rge}(\varphi_e) \text{ is oneindig}\}$

iii) $\{e \mid \varphi_e \text{ is constant (mogelijk partieel)}\} = \{e \mid \varphi_e \text{ neemt hooguit één waarde aan}\}$

iv) $\{j(e,f) \mid W_e \leq_m W_f\}$

v) $\{e \mid W_e \text{ is } m\text{-compleet in } \Sigma_1\}$

Karakteriseer hiervan de eerste drie *volledig*, door te laten zien dat ze m-compleet zijn in hun klasse.

3.2.1 Een exacte klassificatie

In deze paragraaf wil ik een voorbeeld geven van een iets minder eenvoudige exacte klassificatie in de arithmetische hiërarchie. Er is uitgebreide theorie over dit soort klassificaties; zie hiervoor het boek van Soare. Dit gaat echter buiten het bestek van dit diktaat; daarom is onderstaand bewijs met "handen en voeten" gedaan.

Ik neem de verzameling

$$X = \{x \mid \varphi_x \text{ is niet voort te zetten tot een totaal recursieve functie}\}$$

Volgens opgave 43 is $X \neq \emptyset$.

Opgave 69. Bewijs dat X niet Σ_1 is:

a) met de stelling van Myhill-Shepherdson

b) zonder gebruik te maken van de stelling van Myhill-Shepherdson

Met het Tarski-Kuratowski algoritme komen we er gemakkelijk achter dat X een Π_3-verzameling is: er geldt

$$e \in X \Leftrightarrow \forall f(\varphi_f \text{ totaal} \Rightarrow \varphi_e \not\subseteq \varphi_f)$$

De rechterkant van deze equivalentie kunnen we schrijven als

$$\forall f(\forall x \exists y T(1, f, x, y) \to \exists uvw(T(1, e, u, v) \wedge T(1, f, u, w) \wedge U(v) \neq U(w)))$$

oftewel als

$$\forall f \exists uvwx \forall y(T(1, f, x, y) \to (T(1, e, u, v) \wedge T(1, f, u, w) \wedge U(v) \neq U(w)))$$

dus $X \in \Pi_3$.

Propositie 3.13 *X is m-compleet in Π_3.*

Bewijs. Zij A een willekeurige Π_3-verzameling, dus

$$A = \{f \mid \forall x \exists y \forall z R(f, x, y, z)\}$$

voor zeker recursief predikaat R. We moeten een totaal recursieve functie G vinden zodat

$$f \in A \Rightarrow G(f) \in X$$

$$f \notin A \Rightarrow G(f) \notin X$$

Dit gaat als volgt. Volgens de S^m_n-stelling is er een primitief recursieve functie G zodat voor alle f en z, $G(f) \bullet z$ gegeven wordt door de volgende instructies:

> Bereken (zoek) de kleinste W zodat hètzij (1), hètzij (2):
>
> (1) $\exists x \leq j_1(z) \forall y \leq j_2(z) \exists w \leq W \neg R(f, x, y, w)$
> (2) $\forall y \leq j_2(z) \exists w \leq W T(1, j_1(z), j(j_1(z), y), w)$
>
> Zet $G(f) \bullet z$ ongedefinieerd als er niet zo'n W is.
> Als W wel gevonden is, zet $G(f) \bullet z = W$ als (1) zich voordoet.
> Als (1) zich niet voordoet, zoek w met $T(1, j_1(z), z, w)$, en zet $G(f) \bullet z = U(w) + 1$.

Stel nu $f \in A$ dus $\forall x \exists y \forall z R(f, x, y, z)$, en stel φ_u totaal.
Laat voor iedere x, $n_x = \max(\{m \mid \forall y < m \exists z \neg R(f, x, y, z)\})$; en zet

$$N(u) = \max(\{n_x \mid x \leq u\}) + 1$$

Dan is $G(f) \bullet j(u, N(u))$ altijd gedefinieerd (de gezochte W is er op grond van (2), omdat φ_u totaal is), maar de gevonden W kan nooit aan (1) voldoen (ga na); dus $G(f) \bullet j(u, N(u)) = u \bullet j(u, N(u)) + 1$, dus φ_u is geen voortzetting van $\varphi_{G(f)}$. Dit geldt voor elke totale φ_u, dus $G(f) \in X$.

Als $f \notin A$ dan $\exists x \forall y \exists z \neg R(f, x, y, z)$; stel N minimaal zodat $\forall y \exists z \neg R(f, N, y, z)$. Dan geldt:

- Als $j_1(z) \geq N$, is $G(f) \bullet z$ gedefinieerd (W wordt gevonden want (1) zal zich altijd voordoen).
- Als $j_1(z) < N$ hangt $G(f) \bullet z$ af van $j_1(z)$.
  De functie $\lambda n.G(f) \bullet j(j_1(z), n)$ is totaal als $\lambda x.j_1(z) \bullet j(j_1(z), x)$ totaal is, en heeft een eindig domein als $\lambda x.j_1(z) \bullet j(j_1(z), x)$ niet totaal is. In beide gevallen heeft $\lambda n.G(f) \bullet j(j_1(z), n)$ een totaal recursieve voortzetting $F_{j_1(z)}$.

Definieer dus

$$H(z) = \begin{cases} F_0(j_2(z)) & \text{als } j_1(z) = 0 \\ \vdots & \vdots \\ F_{N-1}(j_2(z)) & \text{als } j_1(z) = N - 1 \\ G(f) \bullet z & \text{anders} \end{cases}$$

Dan is H een totaal recursieve voortzetting van $\varphi_{G(f)}$. ∎

3.3 R.e. verzamelingen: recursief, m-compleet, en er tussenin

In deze paragraaf zullen we zien dat het perspectief van many-one reduceerbaarheid een niet-triviale structuur geeft op de collectie van alle r.e. verzamelingen.

Eerst beschouwen we een belangrijke groep r.e. verzamelingen: namelijk die, welke extensioneel zijn als verzameling codes voor r.e. verzamelingen (zie hoofdstuk 2). Deze blijken altijd hetzij recursief, hetzij m-compleet in Σ_1 te zijn.

Vervolgens karakteriseren we de m-complete r.e. verzamelingen: deze zijn "recursief isomorf" met $\mathcal{K}$, de standaard verzameling. Ook zijn het precies de creatieve verzamelingen (Gevolg 3.18 hieronder).

Tenslotte tonen we aan, dat er ook r.e. verzamelingen zijn die noch recursief, noch m-compleet zijn. De structuur van de preordening $(\mathcal{RE}, \leq_m)$ is uiterst ingewikkeld, en onderwerp van veel onderzoek!

3.3.1 Extensionele r.e. verzamelingen

Opgave 70. Laat zien dat uit de stelling van Rice volgt: als $X \subseteq \mathbb{N}$ recursief is en extensioneel als verzameling codes van r.e. verzamelingen, dan $X = \emptyset$ of $X = \mathbb{N}$.

Propositie 3.14 *Als $X \subseteq \mathbb{N}$ r.e. en extensioneel als verzameling codes van r.e. verzamelingen, dan is X hetzij recursief, hetzij m-compleet in Σ_1.*

Bewijs. Stel X voldoet aan de voorwaarden, en $X \neq \emptyset$, $X \neq \mathbb{N}$. Dan volgt uit de opgave hierboven dat X niet recursief is, dus we laten zien dat X m-compleet is in Σ_1.

Nu volgt uit de stelling van Rice-Shapiro, dat als $n \in X$ en $W_n = \emptyset$, dan $X = \mathbb{N}$ (ga na). Dus $W_n = \emptyset \Rightarrow n \notin X$. Omdat $X \neq \emptyset$ is er $e \in X$. Kies zo'n e. Zij $g \in \mathbb{N}$ willekeurig; kies (met behulp van de S^m_n-stelling) een primitief recursieve functie F zodat voor alle x, y:

$$F(x) \bullet y \simeq \mu z.[T(1, e, y, j_1(z)) \wedge T(1, g, x, j_2(z))]$$

Dan geldt:

$$W_{F(x)} = \begin{cases} W_e & \text{als } x \in W_g \\ \emptyset & \text{anders} \end{cases}$$

Oftewel, vanwege de aanname op X: $x \in W_g \Leftrightarrow F(x) \in X$. Dus $W_g \leq_m X$, en daar g willekeurig was is X m-compleet in Σ_1. ∎

3.3.2 m-Complete r.e. verzamelingen

Ter herinnering: een verzameling $X \subseteq \mathbb{N}$ heet creatief als X r.e. is en er een partieel recursieve functie ψ is zodat voor alle e:

$$W_e \cap X = \emptyset \Rightarrow \psi(e) \text{ is gedefinieerd en } \psi(e) \notin W_e \cup X$$

(X heet dan creatief *via* ψ)

Opgave 71. Laat zien dat $\mathcal{K}$ creatief is.

We zullen nu laten zien, dat een r.e. verzameling precies dan m-compleet is in Σ_1, als hij creatief is. Hierbij speelt de volgende stelling een rol, die ook op zichzelf niet oninteressant is; hij wordt wel een "recursief analogon van de stelling van Cantor-Bernstein" genoemd (de stelling van Cantor-Bernstein in de verzamelingenleer zegt, dat als er injectieve functies $X \to Y$ en $Y \to X$ zijn voor verzamelingen X en Y, er ook een bijectie $X \to Y$ is).

Een nieuwe notatie: met $X \leq_1 Y$ wordt bedoeld dat $X = F^{-1}(Y)$ voor zekere totaal recursieve *injectieve* functie F. Ook de relatie $\leq_1$ is een preordening natuurlijk, en er is het begrip "1-compleet in Σ_n", geheel analoog gedefinieerd als m-compleet.

Stelling 3.15 (Isomorfiestelling van Myhill) *Voor $A, B \subseteq \mathbb{N}$: als $A \leq_1 B$ en $B \leq_1 A$ is er een totaal recursieve bijectie $h : \mathbb{N} \to \mathbb{N}$ met $h[A] = B$.*

Bewijs. Laat $A \leq_1 B$ via f en $B \leq_1 A$ via g, dus f, g zijn totaal recursieve 1-1 functies met $A = f^{-1}(B)$ en $B = g^{-1}(A)$. We construeren met primitieve recursie een rij 1-1 functies h_s met eindig domein (strict genomen moet dit gecodeerd worden als eindige rijtjes; dit laat ik aan de lezer over) waaruit ook h_s^{-1} recursief te halen valt. De constructie gaat zo dat $h_s \subseteq h_{s+1}$ voor alle s, en de gezochte functie h is dan $\bigcup_{s\in\mathbb{N}} h_s$.

Zet $h_0 = \emptyset$ (de lege functie). Stel nu h_s gegeven, zódat h_s 1-1 is en h_s^{-1} ook, en voor elke $x \in \mathrm{dom}(h_s)$ geldt $x \in A \Leftrightarrow h_s(x) \in B$.

Als $s + 1 = 2x + 1$ definiëren we $h(x)$. Als reeds $x \in \mathrm{dom}(h_s)$ zetten we $h_{s+1} = h_s$. Anders, bereken de rij $f(x), f(h_s^{-1}(f(x))), \ldots, f(h_s^{-1}f)^n(x), \ldots$, totdat het eerste element y wordt gevonden met $y \notin \mathrm{rge}(h_s)$. Zo'n y moet bestaan omdat h_s 1-1 is en $x \notin \mathrm{dom}(h_s)$. Zet $h_{s+1} = h_s \cup \{(x, y)\}$. Ga na dat h_{s+1} nog steeds 1-1 is en dat voor $n \in \mathrm{dom}(h_{s+1})$: $n \in A$ dan en slechts dan als $h_{s+1}(n) \in B$.

Als $s + 1 = 2x + 2$, definiëren we $h^{-1}(x)$. Dit gaat op dezelfde manier, met de rollen van f, h_s, dom en rge vervangen door g, h_s^{-1}, rge en dom, respectievelijk. ∎

We keren terug tot de creatieve verzamelingen.

Propositie 3.16 *Als X creatief is, is er een injectieve totaal recursieve functie P zodat X creatief is via P.*

Bewijs. Stel X is creatief via φ. Ga na dat er een primitief recursieve G is zodat

$$W_{G(x)} = \begin{cases} W_x & \text{als } \varphi(x) \text{ gedefinieerd} \\ \emptyset & \text{anders} \end{cases}$$

Definieer Q door $Q(x) = \varphi(x)$ of $\varphi(G(x))$, welke het eerst gedefinieerd is (d.w.z. welke met de kortste berekening gedefinieerd is). Dan is Q totaal, want als $\varphi(x)$ niet gedefinieerd is is $W_{G(x)} = \emptyset$, dus is $\varphi(G(x))$ gedefinieerd omdat X creatief is via φ.

Ook geldt: $W_x \cap X = \emptyset \Rightarrow \varphi(x)$ gedefinieerd $\Rightarrow W_x = W_{G(x)}$, dus in dat geval $\varphi(x) \notin X \cup W_x$ èn $\varphi(G(x)) \notin X \cup W_x$. Hieruit volgt dat X ook creatief is via Q, en Q is totaal. Nu moeten we Q nog "omvormen" tot een injectieve functie P:

Laat H een primitief recursieve functie zijn zodat voor alle x geldt:

$$W_{H(x)} = W_x \cup \{Q(x)\}$$

Er volgt dan:

$$W_{H^{n+1}(x)} = W_x \cup \{Q(x), HQ(x), \ldots, H^nQ(x)\}$$

Definieer de functie P als volgt:

- $P(0) = Q(0)$
- Stel $P(0), \ldots, P(x)$ gedefinieerd. Som nu de rij $Q(x+1), QH(x+1), \ldots, QH^n(x+1), \ldots$ op, totdat a) of b):
  a) Zekere $QH^n(x+1) \notin \{P(0), \ldots, P(x)\}$. Zet $P(x+1) = QH^n(x+1)$.
  b) Er treedt een herhaling op: voor zekere n is $QH^{n+1}(x+1) \in \{Q(x+1), \ldots, QH^n(x+1)\}$. Dan geldt $W_{x+1} \cap X \neq \emptyset$ (ga na); zet eenvoudigweg $P(x+1) = \mu y.[y \notin \{P(0), \ldots, P(x)\}]$.

Ga na dat P 1-1 is, totaal, en dat X creatief is via P. ∎

Propositie 3.17 (Myhill) *Als X creatief is, dan $\mathcal{K} \leq_1 X$*

Bewijs. Zij (met propositie 3.16) P injectief en totaal zodat X creatief is via P. Kies een primitief recursieve F zodat

$$W_{F(x,y)} = \begin{cases} \{P(x)\} & \text{als } y \in \mathcal{K} \\ \emptyset & \text{anders} \end{cases}$$

Met de recursiestelling in parameters is er een primitief recursieve injectieve N zodat voor alle y:

$$W_{N(y)} = W_{F(N(y),y)} = \begin{cases} \{P(N(y))\} & \text{als } y \in \mathcal{K} \\ \emptyset & \text{anders} \end{cases}$$

Dan geldt:

$$y \in \mathcal{K} \Rightarrow W_{N(y)} = \{P(N(y))\} \Rightarrow W_{N(y)} \cap X \neq \emptyset \Rightarrow P(N(y)) \in X$$
$$y \notin \mathcal{K} \Rightarrow W_{N(y)} = \emptyset \Rightarrow W_{N(y)} \cap X = \emptyset \Rightarrow P(N(y)) \notin X$$

dus $\mathcal{K} \leq_1 X$ via PN. ■

Opgave 72. Laat zien:

i) Als X creatief is is X strict r.e.;

ii) als X creatief is, is er een oneindige r.e. verzameling disjunct met X;

iii) als X creatief is en Y r.e. en $X \leq_m Y$, dan is Y creatief.

Gevolg 3.18 *Voor $X \subseteq \mathbb{N}$ zijn de volgende uitspraken equivalent:*

i) X is creatief;

ii) X is 1-compleet in Σ_1;

iii) X is m-compleet in Σ_1;

iv) Er is een totaal recursieve bijectie $h : \mathbb{N} \to \mathbb{N}$ met $h[X] = \mathcal{K}$

Opgave 73. Bewijs gevolg 3.18. Gebruik de vorige opgave, propositie 3.17 en stelling 3.15.

Opgave 74. Noem een disjunct paar (A, B) van r.e. verzamelingen *effectief onscheidbaar* als er een partieel recursieve functie ψ is zodat voor alle x en y: als $A \subseteq W_x$ en $B \subseteq W_y$ en $W_x \cap W_y = \emptyset$, dan is $\psi(\langle x, y\rangle)$ gedefinieerd, en $\psi(\langle x, y\rangle) \notin W_x \cup W_y$.

a) Laat zien: we mogen aannemen dat ψ totaal is en 1-1.

b) Laat zien: $A = \{x \mid x \bullet x = 0\}$ en $B = \{x \mid x \bullet x = 1\}$ zijn effectief onscheidbaar.

c) Laat zien: als (A, B) effectief onscheidbaar is, zijn A en B creatief.

Opgave 75. Laat zien dat er disjuncte r.e. verzamelingen $A, B \subseteq \mathbb{N}$ zijn, die wel recursief onscheidbaar, maar niet effectief onscheidbaar zijn.

3.3.3 Simpele r.e. verzamelingen: noch recursief, noch m-compleet

Definitie 3.19 een r.e. verzameling A heet *simpel* als A^c oneindig is maar geen oneindige r.e. deelverzameling bevat (oftewel: A snijdt elke oneindige r.e. verzameling).

Opgave 76. Laat zien: als A een simpele r.e. verzameling is, dan is A niet recursief, en ook niet creatief.

Omdat we weten uit Gevolg 3.18 dat de m-complete verzamelingen precies de creatieve verzamelingen zijn, is het voor ons doel genoeg, aan te tonen dat er simpele verzamelingen bestaan.

Stelling 3.20 (Post, 1944) *Er bestaat een simpele verzameling.*

Bewijs. Definieer de partieel recursieve functie ψ als volgt:

$$\psi(e) \simeq j_1(\mu z.[T(1, e, j_1 z, j_2 z) \wedge j_1 z > 2e])$$

Zij $S = \mathrm{rge}(\psi)$ Dan is S r.e. Omdat altijd $\psi(e) > 2e$ (als $\psi(e)$ gedefinieerd is), bevat voor elke e, S hooguit e elementen uit de verzameling $\{0, 1, \ldots, 2e\}$; en dus is S^c oneindig.

Tenslotte, als $B = W_e$ een oneindige r.e. verzameling is, is $\psi(e)$ gedefinieerd, en $\psi(e) \in S \cap W_e$ (ga na!). Dus S is simpel. ■

3.4 Arithmetische verzamelingen

Een verzameling $A \subseteq \mathbb{N}$ (of $\subseteq \mathbb{N}^k$) heet *arithmetisch* als $A \in \bigcup_{n\in\mathbb{N}} \Sigma_n$. De herkomst van het woord "arithmetisch" is de volgende.

De *taal van de rekenkunde* heeft één constante 0 en drie functieletters: S (1-plaatsig) en P,M (beide 2-plaatsig). De standard-structuur voor deze taal, die we aangeven met $\mathcal{N}$, is de verzameling $\mathbb{N}$ met 0 geïnterpreteerd als 0, en S,P,M respectievelijk als $\lambda x.x + 1$, $\lambda xy.x + y$ en $\lambda xy.xy$. Zoals nog uitvoerig in hoofdstuk 5 aan de orde zal komen geldt het volgende:
A is arithmetisch precies dan als er een formule $\varphi(x)$ met één vrije variabele x in de taal van de rekenkunde is, zodat

$$A = \{n \mid \mathcal{N} \models \varphi[n]\}$$

Uiteraard zijn er maar aftelbaar veel arithmetische verzamelingen. Verzamelingen die niet arithmetisch zijn, kunnen gedefinieerd worden in tweede-orde logica, dat is een formalisme waarin ook kwantoren voorkomen die lopen over deelverzamelingen.

De "tweede-orde definieerbare" deelverzamelingen van $\mathbb{N}$ vormen een hiërarchie die de arithmetische hiërarchie strict bevat; het gedeelte boven de arithmetische verzamelingen heet de *analytische hiërarchie*. Studie van de analytische hiërarchie is van belang in de bewijstheorie en in de descriptieve verzamelingenleer.

4 Relatieve berekenbaarheid en Turing reduceerbaarheid

4.1 Functies partieel recursief in F

De onderverdeling van deelverzamelingen van $\mathbb{N}$ in $\equiv_m$-klassen was vooral belangrijk vanwege het nauwe verband met de arithmetische hiërarchie.

Voor een andere wezenlijke vraag van de recursietheorie voldoet zij echter niet. Die vraag is: welke functies kan ik berekenen, als mij een of andere functie F "gegeven" is? Bijvoorbeeld, als ik de relevante informatie mag "invoeren" in de registermachine?

Oftewel: welke functies G zijn partieel recursief uit F te berekenen?

De volgende definities liggen voor de hand.

Definitie 4.1 De klasse van functies *primitief recursief in* een gegeven functie F is de kleinste klasse functies die $\lambda x.0$, $\lambda \vec{x}.x_i$, $\lambda x.x+1$ en F bevat, en gesloten is onder compositie en primitieve recursie.

Definitie 4.2 De klasse van functies *partieel recursief in* F is de kleinste klasse partiële functies die alle functies primitief recursief in F bevat, en gesloten is onder compositie en minimalisatie.

We kunnen ook een begrip "berekenbaar in F" introduceren, door het concept registermachine uit te breiden. We denken aan een machine die waarden van F langs "magische weg" weet te produceren, wat de naam "orakel" in de volgende definitie verklaart (iets concreter kun je ook denken aan een computer die aangesloten is op een bepaald gegevensbestand dat geraadpleegd mag worden; de vraag hoe dit gegevensbestand tot stand is gekomen, laten we geheel buiten beschouwing).

Definitie 4.3 De programmeertaal voor de *registermachine met orakel* F heeft, behalve de instructies $r_i^+ \Rightarrow j$ en $r_i^- \Rightarrow j, k$ van paragraaf 1.1, ook instructies van de vorm

$$r_i = F(r_j) \Rightarrow k$$

die gelezen worden als: vervang de inhoud van het i-de register door F toegepast op de inhoud van het j-de register (laat alle andere registerinhouden ongewijzigd) en ga naar instructie k.

Coderen we instructie $r_i = F(r_j) \Rightarrow k$ als $\langle i, j, k, 0\rangle$ dan kunnen we paragraaf 1.5 als volgt overschrijven:

- Er is een primitief recursief predikaat $\mathrm{Prog}'(e)$:

$$\mathrm{lh}(e) > 0 \wedge \forall i < \mathrm{lh}(e)[\mathrm{lh}((e)_i) \in \{2,3\} \vee (\mathrm{lh}((e)_i) = 4 \wedge ((e)_i)_3 = 0)]$$

- Met $\varphi^F_{m,e}$ geven we de m-plaatsige partiële functie aan die berekend wordt door het programma dat code e heeft, met orakel F.

- Er is een partiële functie $\Phi^F : \mathbb{N}^3 \rightharpoonup \mathbb{N}$ gedefinieerd door

$$\Phi^F(m,e,x) \simeq \begin{cases} \text{ongedefinieerd} & \text{als } \neg\mathrm{Prog}'(e) \\ \varphi^F_{m,e}(j_1^m(x), \ldots, j_m^m(x)) & \text{anders} \end{cases}$$

We hebben, volledig analoog aan 1.20:

Propositie 4.4 *Φ^F is partieel recursief in F en voor iedere functie $G : \mathbb{N}^k \rightharpoonup \mathbb{N}$ die partieel recursief is in F is er een e zodat*

$$G(x_1, \ldots, x_k) \simeq \Phi^F(k, e, j^k(x_1, \ldots, x_k))$$

voor alle $x_1, \ldots, x_k$.

Het bewijs construeert een predikaat $T^F(m, e, j^m(x_1, \ldots, x_m), y)$; y codeert een berekening op registermachine met orakel F, met programma gecodeerd door e en input $x_1, \ldots, x_m$.

Het predikaat T^F is niet zonder meer primitief recursief, maar primitief recursief *in* F, omdat het moet controleren of de instructies $r_i = F(r_j) \Rightarrow k$ juist zijn uitgevoerd.

De *uitkomstfunctie* $U(y) = ((y)_{\mathrm{lh}(y)-1})_1$ blijft echter dezelfde, en is dus primitief recursief.

Gevolg 4.5 *De klasse functies van de vorm $\varphi^F_{m,e}$ is precies de klasse van functies partieel recursief in F.*

Ook de S^m_n- en recursiestelling hebben we in deze algemenere context:

Propositie 4.6 (S^m_n-stelling) *Er zijn primitief recursieve functies S^m_n, zodat*

$$\varphi^F_{n,S^m_n(e,x_1,\ldots,x_m)}(y_1, \ldots, y_n) \simeq \varphi^F_{m+n,e}(x_1, \ldots, x_m, y_1, \ldots, y_n)$$

Propositie 4.7 (Recursiestelling, primitief recursief uniforme versie) *Er is een primitief recursieve functie T zodat voor alle f en $x_1, \ldots, x_m$:*

$$\varphi^F_{m,T(f)}(x_1, \ldots, x_m) \simeq \varphi^F_{m+1,f}(x_1, \ldots, x_m, T(f))$$

De reden dat de functies S^m_n primitief recursief blijven (je zou misschien op het eerste gezicht verwachten: primitief recursief in F) is dat zij manipuleren op codes van programma's en niet op berekeningen, en de code van een programma met orakel F hangt niet van F af. Tenslotte is de functie T uit de recursiestelling een combinatie van S^m_n-functies (zie het bewijs van stelling 1.24).

Opgave 77. Gebruik het feit (dat wel het *Use Principle* wordt genoemd), dat elke berekening van $\varphi^F_{m,e}(x_1, \ldots, x_m)$ maar eindig veel waarden van F gebruikt, om te laten zien dat de partiële functie

$$\Phi_{e,x_1,\ldots,x_m} : \mathbb{N}^{\mathbb{N}} \rightharpoonup \mathbb{N}$$

gedefinieerd door $\Phi_{e,x_1,\ldots,x_m}(F) \simeq \varphi^F_{m,e}(x_1, \ldots, x_m)$, continu is op zijn domein (Geef $\mathbb{N}^{\mathbb{N}}$ dezelfde topologie als in paragraaf 2.3).

Bij het definiëren van functies partieel recursief in F schrijf je dezelfde soort vergelijkingen op als bij gewone partieel recursieve functies, alleen mag je F daarbij gebruiken alsof hij recursief is.

De volgende overweging is ook nuttig: definieer een predikaat

$$T^\sigma(m, e, j^m(x_1, \ldots, x_m), y)$$

door: y codeert een berekening "met orakel σ" voor een (code van) een eindig rijtje σ; d.w.z. elke keer als $r_i = F(r_j)$ wordt gevraagd geldt $r_j < \mathrm{lh}(\sigma)$ en wordt r_i vervangen door $(\sigma)_{r_j}$.

Dit predikaat is natuurlijk primitief recursief. We kunnen nu in elke berekening die partieel recursief in F is, het partieel recursieve deel scheiden van het orakel:

Propositie 4.8 *$\varphi^F_{m,e}(x_1, \ldots, x_m) = y$ is equivalent met*

$$\exists\sigma[\forall i < \mathrm{lh}(\sigma)(\sigma)_i = F(i) \wedge \exists w(T^\sigma(m, e, j^m(x_1, \ldots, x_m), w) \wedge U(w) = y)]$$

Dus afgezien van de kwantor $\exists\sigma$ staat hier een conjunctie van iets primitief recursiefs in F en iets recursief opsombaars.

Definitie 4.9 Voor totale functies F en G zeggen we dat F *Turing reduceerbaar is tot* G, notatie $F \leq_T G$, als F recursief is in G. Voor deelverzamelingen A en B van $\mathbb{N}$ ($\mathbb{N}^k$) zeggen we dat A Turing reduceerbaar is tot B of $A \leq_T B$ als $\chi_A \leq_T \chi_B$.

Opgave 78. Bewijs dat de relatie $\leq_T$ een preordening is op $\mathcal{P}(\mathbb{N})$.

Opgave 79.Bewijs:

$$\begin{gathered} A \leq_m B \Rightarrow A \leq_T B \\ A^c \leq_T A \\ A \leq_T B \not\Rightarrow A \leq_m B \end{gathered}$$

Net als bij $\leq_m$ kunnen we $\mathcal{P}(\mathbb{N})$ opdelen in $\equiv_T$-equivalentieklassen. De $\equiv_T$-klasse van $A \subseteq \mathbb{N}$ wordt ook wel de *graad van onoplosbaarheid* van A genoemd.

Opgave 80. Bewijs dat elke $A, B \subseteq \mathbb{N}$ een supremum hebben m.b.t. $\leq_T$.

Opgave 81. Bewijs, dat niet geldt: als $A \leq_T B$ en B is r.e., dan is ook A r.e.

Opgave 82. Gegeven A en B, bewijs dat de volgende uitspraken equivalent zijn:

i) $B \leq_T A$

ii) er zijn totaal recursieve functies f en g zodat:

$$\begin{aligned} x \in B \quad &\Leftrightarrow \quad \exists\sigma(\sigma \in W_{f(x)} \wedge \forall i < \mathrm{lh}(\sigma)(\sigma)_i = \chi_A(i)) \\ x \notin B \quad &\Leftrightarrow \quad \exists\sigma(\sigma \in W_{g(x)} \wedge \forall i < \mathrm{lh}(\sigma)(\sigma)_i = \chi_A(i)) \end{aligned}$$

(Aanwijzing: gebruik propositie 4.8)

4.2 Verzamelingen r.e. in F; de jump-operatie

Ook een belangrijk deel van hoofdstuk 2 kan worden herschreven in de context van "recursief in F" i.p.v. gewoon recursief. In het bijzonder hebben we:

Propositie 4.10 *Laat $\mathcal{K}^F = \{e \mid \varphi^F_{1,e}(e)$ is gedefinieerd$\}$. Dan is $\mathcal{K}^F$ niet recursief in F.*

Het bewijs is analoog aan het standaard-geval.

Definitie 4.11 A heet *r.e. in F* als $A = \mathrm{dom}(\varphi^F_{1,e})$ voor zekere e (equivalent: één van de andere condities in propositie 2.4, geschikt gerelativeerd naar F). A heet r.e. in $B \subseteq \mathbb{N}$ als A r.e. is in χ_B.

Definitie 4.12 Voor een verzameling A heet $\mathcal{K}^{\chi_A}$ de *jump* van A; we noteren deze verzameling als A'.

De operatie $A \mapsto A'$ behoudt $\equiv_T$-equivalentie, zoals we in de volgende propositie zullen zien, en kan dus worden gedefinieerd op $\equiv_T$-equivalentieklassen. Het is een hulpmiddel bij het bestuderen van de partiële ordening op de $\equiv_T$-klassen.

Propositie 4.13 (Jump-stelling)

i) *A' is r.e. in A*

ii) $A' \not\leq_T A$

iii) *B r.e. in $A \Leftrightarrow B \leq_m A'$*

iv) *A r.e. in B, $B \leq_T C \Rightarrow A$ r.e. in C*

v) $B \leq_T A \Leftrightarrow B' \leq_m A'$

vi) *A r.e. in $B \Leftrightarrow A$ r.e. in B^c*

Bewijs. i) is evident.
ii) is propositie 4.10.
iii)$\Rightarrow$ volgt door geschikte relativering van de bewering dat $\mathcal{K}$ m-compleet is in Σ_1, en $\Leftarrow$ uit i) en de bewering dat als A r.e. is in C en $B \leq_m A$, dan B r.e. in C.
Voor iv): als A van de vorm is $\{x \mid \exists y P(x,y)\}$ met P recursief in B, dan is P ook recursief in C, dus is A r.e. in C.
v)$\Rightarrow$ volgt uit iii), want $B \leq_T A$ impliceert dat B' r.e. is in A. Voor $\Leftarrow$, relativeer propositie 2.6i) tot: $B \leq_T A$ geldt precies dan als zowel B als B^c r.e. in A zijn. Uit iii) volgt tevens $B \leq_m B'$ en $B^c \leq_m B'$. Nu volgt: als $B' \leq_m A'$ dan is B r.e. in A en B^c r.e. in A, dus $B \leq_T A$.
Tenslotte volgt vi) uit opgave 79. ∎

Propositie 4.13 legt niet alleen verband tussen de begrippen $\leq_m$ en $\leq_T$; er volgt ook meteen (uit v) en opgave 79), dat $B \leq_T A$ impliceert, dat $B' \leq_T A'$ (en dus ook: als $B \equiv_T A$ dan $B' \equiv_T A'$). De afbeelding $(-)'$ is dus goed gedefinieerd op $\equiv_T$-klassen, en is *monotoon*

In dit diktaat zullen we voor $\equiv_T$-klassen griekse letters gebruiken: $\alpha, \beta, \ldots$

De $\equiv_T$-klasse van $\emptyset$ noteren we met $\mathbb{O}$; zij bestaat precies uit de recursieve verzamelingen. Het is duidelijk dat $\mathbb{O}$ het kleinste element is (m.b.t. $\leq_T$) van de verzameling van $\equiv_T$-klassen. De structuur van de partiële ordening van $\equiv_T$-klassen blijkt zeer ingewikkeld te zijn; veel geavanceerde recursietheorie (de zg. "degree theory") gaat hierover.

Opgave 83. Bewijs dat $\mathcal{K}' \in \Sigma_2$.

Schrijven we $A^{(n)}$ voor de n maal herhaalde jump van A, dan geldt in het algemeen: $\emptyset^{(n)}$ is m-compleet in Σ_n.

In het begin van de jaren '40 van de twintigste eeuw stelde E. Post de vraag, of elke r.e. verzameling hetzij recursief, hetzij $\equiv_T$-equivalent met $\mathcal{K}$ is; anders gezegd, of elke r.e. verzameling een element is van $\mathbb{O} \cup \mathbb{O}'$.

Dit stond een tijdlang bekend als "het probleem van Post". In 1956/57 beantwoordden Friedberg en Muchnik onafhankelijk van elkaar deze vraag ontkennend: er is een r.e. verzameling A zodat $\emptyset <_T A <_T \mathcal{K}$ ($<_T$ duidt aan: $\leq_T$ en $\not\equiv_T$). Ook zijn er r.e. verzamelingen A en B waarvan de $\equiv_T$-klassen α en β onvergelijkbaar zijn m.b.t. $\leq_T$. Voor zulke A en B geldt dan natuurlijk ook $\emptyset <_T A <_T \mathcal{K}$ en $\emptyset <_T B <_T \mathcal{K}$.

Opgave 84. Bewijs deze laatste zin.

Aangezien we weten, dat uit $A \leq_m B$ volgt, dat $A \leq_T B$, hebben we hiermee nog een bewijs, dat er r.e. verzamelingen zijn die noch recursief, noch m-compleet in Σ_1 zijn. De stelling van Friedberg en Muchnik is echter wezenlijk sterker (en moeilijker) dan stelling 3.20.

Ook zien we, dat er $\equiv_T$-klassen α zijn die niet van de vorm β' zijn: de jump-operatie is niet surjectief. Tot slot van deze paragraaf behandel ik een stelling die zegt dat de jump-operatie wèl surjectief is op de $\equiv_T$-klassen $\geq \mathbb{O}'$:

Stelling 4.14 (Friedberg) *Stel $\beta \geq \mathbb{O}'$. dan is er een α zodat $\alpha' = \alpha \sqcup \mathbb{O}' = \beta$.*

Bewijs. Kies $B \in \beta$. Omdat $\beta \geq \mathbb{O}'$ is elke r.e. verzameling recursief in B. We definiëren nu een B-recursieve opsomming van rijtjes $(f_s | s \in \mathbb{N})$, zodat f_s een beginsegment is van f_t (notatie: $f_s \preceq f_t$) als $s \leq t$, als volgt:
$f_0 = \langle\rangle$, het lege rijtje. Stel f_s is gedefinieerd.

- Als s even, $s = 2e$, kijk of

$$\exists\sigma\exists t[f_s \preceq \sigma \wedge T^\sigma(1, e, e, t) \wedge \forall i < \mathrm{lh}(\sigma)(\sigma)_i \in \{0,1\}] \quad (*)$$

  Als er een σ is die aan $(*)$ voldoet, laat f_{s+1} de minimale zulke σ; anders, zet $f_{s+1} = f_s$.

- Als s oneven, $s = 2e+1$; zet $f_{s+1} = f_s \star \langle\chi_B(e)\rangle$.

Merk op dat $f_s \preceq f_{s+1}$ en dat de lengte van f_s groeit bij elke oneven s, dus de collectie van alle f_s bepaalt een (totale) functie f.

Omdat conditie $(*)$ r.e. is dus recursief in $\mathbb{O}'$, dus recursief in B, en bij oneven s alleen χ_B wordt gebruikt, is de functie f recursief in B.

Laat $A = \{x \mid f(x) = 0\}$ en α de $\equiv_T$-klasse van A. Dan is $A \leq_T B$, dus $\alpha \sqcup \mathbb{O}' \leq_T \beta$ is duidelijk. Ook geldt altijd $\alpha \sqcup \mathbb{O}' \leq_T \alpha'$ (ga na). We zijn dus klaar als we hebben laten zien:

1) $\alpha' \leq_T \beta$

2) $\beta \leq_T \alpha \sqcup \mathbb{O}'$

Ad 1): om te beslissen of $e \in A' = \{e \mid \varphi^A_{1,e}(e)$ is gedefinieerd$\}$, kijk of $(*)$ geldt voor $s = 2e$. Als $(*)$ geldt is er dus een minimaal 0-1 rijtje σ met $f_{2e} \preceq \sigma$ en $T^\sigma(1, e, e, t)$ voor zekere t; maar dan geldt per definitie, dat $e \in A'$.

Als $(*)$ niet geldt voor $s = 2e$ dan is het duidelijk dat $e \notin A'$. Omdat conditie $(*)$ recursief is in $\mathbb{O}'$ en f_s recursief is in B, volgt $A' \leq_T B$, en dus $\alpha' \leq \beta$.

Ad 2): $\chi_B(e)$ is de laatste component van het rijtje f_{2e+2} dus B is recursief in de opsomming $(f_s | s \in \mathbb{N})$.

Maar f_{2e+1} wordt recursief in $\mathbb{O}'$ uit f_{2e} verkregen en f_{2e+2} wordt recursief in A uit f_{2e+1} verkregen (want $f = \chi_A$, dus $f_{2e+2} = f_{2e+1} \star \langle \chi_A(\mathrm{lh}(f_{2e+1}))\rangle$), dus de opsomming $(f_s | s \in \mathbb{N})$ is recursief in $A \sqcup \emptyset'$, oftewel $B \leq_T A \sqcup \emptyset'$, dus $\beta \leq \alpha \sqcup \mathbb{O}'$. ■

4.3 De gerelativeerde arithmetische hiërarchie

Volkomen rechttoe rechtaan is de definitie van Σ^F_n, Π^F_n: vervang in de definitie van Σ_n resp. Π_n steeds recursief door recursief in F, en het T-predikaat door T^F.

Alle stellingen van paragraaf 3.2 generaliseren: er zijn Kleene-normaalvormen voor Σ^F_n, die m-compleet zijn in Σ^F_n; de hiërarchie is strict, d.w.z. klapt niet in elkaar, etc.

We noemen een verzameling A *arithmetisch in* F als $A \in \bigcup_{n\in\mathbb{N}} \Sigma^F_n$. De deelverzamelingen van $\mathbb{N}$ die arithmetisch in F zijn, kunnen we ook zo voorstellen: voeg aan de taal van de rekenkunde (zie paragraaf 3.4) een nieuw functiesymbool $\overline{F}$ toe, en interpreteer dat als F in de standaardstructuur $\mathcal{N}$. Weer geldt: A is arithmetisch in F als er een formule $\varphi(x)$ in de uitgebreide taal is met één vrije variabele x, zodat

$$A = \{n \in \mathbb{N} \mid \mathcal{N} \models \varphi[n]\}$$

Nog een laatste opgave:

Opgave 85. Bewijs voor alle $n \geq 1$:

$$X \in \Sigma^Y_n \Leftrightarrow X \leq_m Y^{(n)}$$

Aanwijzing: gebruik inductie.

5 Stellingen van Gödel, Church en Tarski

5.1 Inleiding

In de eerste decennia van deze eeuw woedde een felle discussie binnen de wiskunde, over de grondslagen van het vak. Een aantal wiskundigen, onder wie de Nederlander L.E.J. Brouwer, begonnen openlijk te twijfelen aan de correctheid van bepaalde redeneerprincipes: met name het keuze-axioma en het principe van het "uitgesloten derde" moesten het ontgelden. Brouwer was verder van mening dat over oneindige wiskundige objecten (zoals in wezen: reële getallen, functies: $\mathbb{N} \to \mathbb{N}$ etc.) niet gepraat mocht worden alsof zij "bestaan"; zulk soort objecten moest gezien worden als immer onvoltooide constructieprocessen, en omdat een reëel getal dus altijd "in wording" is (bijvoorbeeld als fundamentaalrij) kan ik niet beweren dat het hètzij groter dan 0, hètzij kleiner dan 0, hètzij gelijk aan 0 is.

Brouwer had gezag, omdat hij in de eerste 15 jaar van de eeuw een paar moeilijke problemen in het toen nog gloednieuwe vak topologie had opgelost (zoals zijn dekpuntsstelling, de invariantie van dimensie –i.e. $\mathbb{R}^n$ is niet homeomorf met $\mathbb{R}^m$ als $m \neq n$– en zijn bewijs van de stelling van Jordan; tegenwoordig zijn deze stellingen eenvoudige gevolgen van de algebraïsche topologie geworden). Hij riep echter uiteraard ook weerstand op en vond de grootste wiskundige van zijn tijd, David Hilbert, tegenover zich. Het debat wordt vaak (ook in Engelse teksten!) aangeduid met het Duitse woord "Grundlagenstreit".

Hilbert stelde zich ten doel Brouwers ketterijen te weerleggen door de correctheid van het gangbare wiskundige redeneren aan te tonen met algemeen aanvaarde, constructieve middelen (hoewel het uitgesloten derde was toegestaan). Een vaak aangehaalde uitspraak van hem is: "Aus dem Paradis, das Cantor uns geschaffen, soll uns keiner vertreiben können". Zijn streven staat bekend als "Hilberts programma".

De onmiskenbare verdienste van Hilberts programma is het geweest, dat het een belangrijke stoot heeft gegeven aan het vak logica. In feite ontstond in de jaren twintig het vak "bewijstheorie", dat de combinatorische eigenschappen van bewijzen in de predikatenlogica of andere systemen onderzoekt. Er ontstond een hele school van Duitse logici (Ackermann, Gödel, Bernays, Gentzen,...) en nog altijd is de Duitse logica sterk bewijstheoretisch georiënteerd.

Gödels onvolledigheidsresultaten torpedeerden Hilberts programma echter volledig. Laten we even zien waarom:

Tot de "constructieve middelen" die Hilbert wilde kunnen gebruiken, hoorden toch minstens: de constructie van de natuurlijke getallen en de elementaire eigenschappen van optelling en vermenigvuldiging, alsmede het principe van volledige inductie over natuurlijke getallen.

Maar, zo liet Gödel zien, in een taal in de predikatenlogica die over natuurlijke getallen kan praten kunnen we, door codes toe te kennen aan formules en bewijzen, over het logische systeem zèlf praten. Als nu T een theorie in die taal is die de elementaire eigenschappen van optelling en vermenigvuldiging bewijst, die inductie kent en die een beslisbare verzameling axioma's heeft, dan is de bewering dat T consistent is uit te drukken door een zin in de taal van T, en er geldt: als T consistent is bewijst T niet de zin, die uitdrukt dat T consistent is.

Dus met al te simpele middelen kan niet worden bewezen dat de wiskunde consistent is.

Hilbert slaagde dus niet in zijn opzet, hoewel iedere derdejaars student wiskunde wel kan zien, dat hij historisch gezien het debat met Brouwer glansrijk heeft gewonnen.

Het "intuïtionisme", Brouwers wiskunde zonder uitgesloten derde en keuze-axioma, is buitengewoon interessant, vooral vanuit logisch standpunt, en wordt nog steeds bestudeerd. Maar als ideologie heeft het weinig aanhangers meer.

5.2 Het formele axiomasysteem van de rekenkunde: PA

Definitie 5.1 *De* taal van de rekenkunde *heeft 4 niet-logische tekens: een constante 0, een éénplaatsig functiesymbool S (voor "opvolger") en twee tweeplaatsige functiesymbolen $+$ en $\cdot$ ("plus" en "maal"). We noemen deze taal $\mathcal{L}_{PA}$.*

Definitie 5.2 *De* Peano rekenkunde, *ook wel aangeduid met* PA *(van: Peano Arithmetic) is de* $\mathcal{L}_{PA}$*-theorie geaxiomatiseerd door de volgende axioma's:*

(1) $\forall x \neg (S(x) = 0)$

(2) $\forall x \forall y (S(x) = S(y) \to x = y)$

(3) *a)* $x + 0 = x$
b) $x + S(y) = S(x + y)$

(4) *a)* $x \cdot 0 = 0$
b) $x \cdot S(y) = x \cdot y + x$

(5) Voor elke formule φ *met vrije variabele* x *een axioma* $(5)_\varphi$*:*

$$\varphi[0/x] \wedge \forall x(\varphi \to \varphi[S(x)/x]) \to \forall x \varphi$$

De axioma's $(5)_\varphi$ heten *inductie-axioma's*. Ze hebben allemaal dezelfde vorm. Zo'n vorm waarin voor φ een willekeurige formule kan worden ingevuld noemt men een (axioma-)*schema*.

We moeten nu van een aantal $\mathcal{L}_{PA}$-zinnen ψ laten zien dat $PA \vdash \psi$. Dit schrijven we niet altijd precies met natuurlijke deductie uit, maar de redeneringen die we geven, kunnen natuurlijk wel in natuurlijke deductie-vorm worden uitgeschreven. Een voorbeeld:

Propositie 5.3

i) $PA \vdash \forall x(x = 0 \vee \exists y(x = S(y)))$

ii) $PA \vdash \forall x \forall y \forall z(x + (y + z) = (x + y) + z)$

iii) $PA \vdash \forall x \forall y(x + y = y + x)$

Bewijs. i) Laat $\varphi(x)$ de formule $x = 0 \vee \exists y(x = S(y))$ zijn. Dan geldt $PA \vdash \varphi[0/x]$ en $PA \vdash \forall x \varphi[S(x)/x]$ (ga na); met $(5)_\varphi$ volgt $PA \vdash \forall x \varphi$.
ii) Laat $\varphi(z)$ de formule $\forall x \forall y(x + (y + z) = (x + y) + z)$ zijn. Omdat $PA \vdash x + y = (x + y) + 0$ en $PA \vdash x + y = x + (y + 0)$ wegens (3)a), en

$$PA \vdash (x + y) + z = x + (y + z) \to S((x + y) + z) = S(x + (y + z))$$

volgt met axioma's (3)a) en (3)b): $PA \vdash \varphi(z) \to \varphi[S(z)/z]$, dus $PA \vdash \forall z \varphi$ met $(5)_\varphi$.
iii) Opgave. ∎

Opgave 86. Bewijs propositie 5.3iii)

De theorie PA heeft duidelijk een model $\mathcal{N}$ met als onderliggende verzameling $\mathbb{N}$, en zodat $0^{\mathcal{N}} = 0$, $S^{\mathcal{N}} = \lambda x.x + 1$, $+^{\mathcal{N}} = +$, $\cdot^{\mathcal{N}} = \cdot$. We noemen dit het *standaardmodel*.

Als φ precies de vrije variabelen $x_1, \ldots, x_n$ bevat en $k_1, \ldots, k_n \in \mathbb{N}$, schrijven we $\mathcal{N} \models \varphi[k_1, \ldots, k_n]$ voor de uitspraak dat φ waar is in $\mathcal{N}$ onder de valuatie die k_i aan x_i toekent voor $1 \leq i \leq n$. Bijvoorbeeld als φ de formule $x + y = z$ is, is $\mathcal{N} \models \varphi[1, 15, 38]$ de uitspraak dat 1+15=38.

In principe zijn er voor elk natuurlijk getal n oneindig veel gesloten $\mathcal{L}_{PA}$-termen t zodat $t^{\mathcal{N}} = n$. Wij zonderen nu speciale termen uit:

Definitie 5.4 *Voor elke* $n \in \mathbb{N}$ *is* $\overline{n}$ *de term* $S^n(0)$*. Termen van deze vorm noemen we* nummers.

Let op het verschil tussen de uitdrukkingen $\mathcal{N} \models \varphi[k_1, \ldots, k_n]$ en $\mathcal{N} \models \varphi[\overline{k_1}/x_1, \ldots, \overline{k_n}/x_n]$! $\varphi[\overline{k_1}/x_1, \ldots, \overline{k_n}/x_n]$ is een $\mathcal{L}_{PA}$-formule, maar $\varphi[k_1, \ldots, k_n]$ niet.

Definitie 5.5 *Een k-plaatsige relatie $R \subseteq \mathbb{N}^k$ heet* nummersgewijs representeerbaar *als er een $\mathcal{L}_{PA}$-formule φ_R met k vrije variabelen $x_1, \ldots, x_k$ is zódat voor alle $n_1, \ldots, n_k \in \mathbb{N}$:*

$$R(n_1, \ldots, n_k) \Rightarrow PA \vdash \varphi_R[\overline{n_1}/x_1, \ldots, \overline{n_k}/x_k] \text{ en}$$

$$\neg R(n_1, \ldots, n_k) \Rightarrow PA \vdash \neg\varphi_R[\overline{n_1}/x_1, \ldots, \overline{n_k}/x_k]$$

Een k-plaatsige functie $F : \mathbb{N}^k \to \mathbb{N}$ heet nummersgewijs representeerbaar als er een $\mathcal{L}_{PA}$-formule φ_F met $k+1$ vrije variabelen $x_1, \ldots, x_{k+1}$ is zodat voor alle $n_1, \ldots, n_{k+1} \in \mathbb{N}$:

$$F(n_1, \ldots, n_k) = n_{k+1} \Rightarrow PA \vdash \varphi_F[\overline{n_1}/x_1, \ldots, \overline{n_{k+1}}/x_{k+1}] \text{ en}$$

$$PA \vdash \forall x_1, \ldots, x_k \forall v \forall w(\varphi_F(\vec{x}, v) \wedge \varphi_F(\vec{x}, w) \to v = w)$$

We zeggen dat φ_R en φ_F R resp. F representeren.

Opgave 87. Bewijs: als $F : \mathbb{N}^k \to \mathbb{N}$ nummersgewijs representeerbaar is, dan is de grafiek van F ook nummersgewijs representeerbaar als $k+1$-plaatsige relatie.

Opgave 88. Als $F : \mathbb{N}^k \to \mathbb{N}$ nummersgewijs representeerbaar is, dan is er ook een $\mathcal{L}_{PA}$-formule ψ zodat:

i) ψ representeert F nummersgewijs;

ii) $PA \vdash \forall x_1 \ldots \forall x_k \exists y \psi(x_1, \ldots, x_k, y)$

Propositie 5.6 *De relaties $R = \{(x, y) \mid x < y\}$ en $S = \{(x, y) \mid x \leq y\}$ zijn nummersgewijs representeerbaar.*

Opgave 89. Bewijs propositie 5.6 (Hint: neem voor φ_R bijv. de formule $\exists z(x + S(z) = y)$)

Opgave 90. Laat φ_R de formule $\exists z(x + S(z) = y)$ zijn. Laat zien dat "PA bewijst dat $<$ een lineaire ordening is", i.e. :

i) $PA \vdash \neg\varphi_R(x, x)$

ii) $PA \vdash \varphi_R(x, y) \wedge \varphi_R(y, z) \to \varphi_R(x, z)$

iii) $PA \vdash \varphi(x, y) \vee x = y \vee \varphi(y, x)$

Opgave 91. φ_R als boven. Laat zien dat "PA bewijst dat $<$ een welordening is", in de zin dat voor elke formule ψ met vrije variabele x:

$$PA \vdash \exists x \psi \to \exists x(\psi \wedge \forall y(\varphi_R(y, x) \to \neg\psi(y)))$$

Opgave 92. Laat zien dat de functie $\lambda xy.$(rest van x bij deling door y) nummersgewijs representeerbaar is.

5.3 Representatie van primitief recursieve functies in PA

In deze paragraaf laten we zien dat iedere primitief recursieve functie nummersgewijs representeerbaar is. Moeilijkheid hierbij is natuurlijk dat we in het schema voor primitieve recursie:

$$\begin{array}{rcl} F(0, \vec{x}) & = & G(\vec{x}) \\ F(y+1, \vec{x}) & = & H(y, F(y, \vec{x}), \vec{x}) \end{array}$$

het symbool F op twee manieren gebruiken.

"**Opgave 93**." Probeer eens te laten zien, dat $\lambda xy.x^y$ nummersgewijs representeerbaar is.

Opgave 94. Laat zien dat de functies $\lambda\vec{x}.0$, $\lambda x_1 \cdots x_k.x_i$ $(1 \leq i \leq k)$ en $\lambda x.x+1$ nummersgewijs representeerbaar zijn, en dat als $G, H_1, \ldots, H_p$ nummersgewijs representeerbaar zijn, de functie $\lambda\vec{x}.G(H_1(\vec{x}), \ldots, H_p(\vec{x}))$ het ook is.

Het idee voor de representatie van primitieve recursie is om rijtjescodering te gebruiken. Stel dat in het definitieschema

$$\begin{array}{rcl} F(0,\vec{x}) & = & G(\vec{x}) \\ F(y+1,\vec{x}) & = & H(y, F(y,\vec{x}), \vec{x}) \end{array}$$

G en H gerepresenteerd worden door φ_G en φ_H respectievelijk; als we de terminologie van rijtjes konden gebruiken, konden we F representeren door de volgende formule $\varphi_F(y, \vec{x}, z)$:

$$\exists v(\mathrm{lh}(v) = y+1 \wedge \varphi_G(\vec{x}, (v)_0) \wedge \forall i < y \varphi_H(i, (v)_i, \vec{x}, (v)_{i+1}) \wedge z = (v)_y)$$

Gödels eerste vondst was een primitieve manier van rijtjescodering, die onmiddellijk representeerbaar is in PA.

Lemma 5.7 (Gödels β-functie) *Er is een primitief recursieve functie $\beta(c,d,i)$ van 3 variabelen, zodat:*

i) *β is nummersgewijs representeerbaar;*

ii) *voor elk $n+1$-tal $a_0, \ldots, a_n \in \mathbb{N}$ zijn er c en d zodat voor alle i met $0 \leq i \leq n$: $\beta(c,d,i) = a_i$*

Bewijs. Definieer β als:

$$\beta(c,d,i) = \text{rest van } c \text{ bij deling door } (i+1)d+1$$

i): volgt meteen uit opgave 92.
ii): Laat $a_0, \ldots, a_n$ gegeven. Zij $s = \max(\{n, a_0, \ldots, a_n\})$ en zet $d = s!$.

Dan zijn voor $0 \leq i < j \leq n$ de getallen $(i+1)d+1$ en $(j+1)d+1$ onderling ondeelbaar, want als p een gemeenschappelijke priemfactor zou zijn, dan ook $p|(j-i)d$; maar zowel $(j-i)$ als d is een deler van $s!$ en het is duidelijk dat $(i+1)d+1 = (i+1)s!+1$ geen deler met $s!$ gemeen heeft.

Omdat nu de getallen $d+1, 2d+1, \ldots, (n+1)d+1$ onderling ondeelbaar zijn en $> a_0, \ldots, a_n$, volgt uit de Chinese reststelling dat er (modulo het produkt uniek, maar dat is niet van belang) een c is zodat de rest van c bij deling door $(i+1)d+1$ gelijk is aan a_i, oftewel $\beta(c,d,i) = a_i$. ∎

Stelling 5.8 *Elke primitief recursieve functie is nummersgewijs representeerbaar.*

Bewijs. Wie opgave 94 gemaakt heeft weet, dat de nummersgewijs representeerbare functies de basisfuncties bevatten en gesloten zijn onder compositie, zodat resteert te bewijzen dat ze gesloten zijn onder primitieve recursie.

Stel F is gedefinieerd uit G en H met primitieve recursie, en φ_G en φ_H representeren G en H. Laat B een formule met 4 vrije variabelen zijn die de functie β van lemma 5.7 representeert. Stel F is $k+1$-plaatsig, en φ_R representeert de $<$-relatie.

Laat $\varphi_F(y, x_1, \ldots, x_k, v)$ de volgende formule zijn:

$$\exists c \exists d((1) \wedge (2) \wedge (3))$$

waar

(1) $\exists a(B(c,d,0,a) \wedge \varphi_G(x_1, \ldots, x_k, a))$

(2) $\forall i(\varphi_R(i,y) \to \exists b \exists b'(B(c,d,i,b) \wedge \varphi_H(i,b,x_1,\ldots,x_k,b') \wedge B(c,d,S(i),b')))$

(3) $B(c,d,y,v)$

Informeel zegt dit, als we a_i schrijven voor $\beta(c,d,i)$:

$$a_0 = G(\vec{x}) \wedge \forall i < y(H(i, a_i, \vec{x}) = a_{i+1}) \wedge v = a_y$$

We moeten nu voor φ_F bewijzen:

i) Voor alle $n_1, \ldots, n_{k+2}$: $F(n_1, \ldots, n_{k+1}) = n_{k+2} \Rightarrow PA \vdash \varphi_F(\overline{n_1}, \ldots, \overline{n_{k+2}})$

ii) $PA \vdash \forall x_1 \cdots x_{k+1} \forall v \forall w(\varphi_F(\vec{x}, v) \wedge \varphi_F(\vec{x}, w) \rightarrow v = w)$

Voor i), laat $a_0, \ldots, a_{n_1}$ resp. $F(0, n_2, \ldots, n_{k+1}), \ldots, F(n_1, \ldots, n_{k+1})$ zijn. Kies c en d voor $a_0, \ldots, a_{n_1}$ als in lemma 5.7ii). Uit de definitie van F en de representerende eigenschappen van φ_G,φ_H en B volgt dan onmiddellijk dat (1), (2) en (3) bewijsbaar zijn voor $\overline{c}, \overline{d}, \overline{n_1}, \ldots, \overline{n_{k+2}}$ (voor (2) moet je laten zien dat voor elk natuurlijk getal n:

$$PA \vdash \forall i(\varphi_R(i, \overline{n}) \rightarrow i = \overline{0} \vee \cdots \vee i = \overline{n-1})$$

Dit laat ik aan de lezer over (inductie naar n)). Door $\exists$-introductie volgt $\varphi_F(\overline{n_1}, \ldots, \overline{n_{k+2}})$.
ii) volgt met inductie in PA (inductie "naar y"):

$$PA \vdash \varphi_F(0, x_1, \ldots, x_k, v) \wedge \varphi_F(0, x_1, \ldots, x_k, w) \rightarrow v = w$$

want ga zelf na dat $PA \vdash \varphi_F(0, x_1, \ldots, x_k, v) \rightarrow \varphi_G(x_1, \ldots, x_k, v)$, en gebruik dat φ_G G representeert.

Stel nu $\varphi_F(S(y), x_1, \ldots, x_k, w) \wedge \varphi_F(S(y), x_1, \ldots, x_k, w')$. Uit de definitie van φ_F volgt meteen:

$$\exists v(\varphi_F(y, x_1, \ldots, x_k, v) \wedge \varphi_H(y, v, x_1, \ldots, x_k, w))$$
$$\exists v'(\varphi_F(y, x_1, \ldots, x_k, v') \wedge \varphi_H(y, v', x_1, \ldots, x_k, w'))$$

Gegeven zulke v en v', volgt met de "inductiehypothese" dat $v = v'$, zodat $w = w'$ volgt vanwege de representerende eigenschap van φ_H. ■

Het laatste stukje van het bewijs was een voorbeeld van "redeneren in PA". Het is in feite een informele redenering, waarbij we echter alleen maar principes gebruiken die in PA afleidbaar zijn, zodat we weten dat de hele bewijsgang ook formeel kan worden uitgevoerd.

Als we naar de definitie van de representerende formule φ_F in het bewijs van stelling 5.8 kijken, zien we dat deze, behalve φ_G, φ_H en B, één universele kwantor $\forall i$ bevat en verder alleen $\exists$-kwantoren; de kwantor $\forall i$ komt voor in de combinatie $\forall i(\varphi_R(i, y) \rightarrow \cdots)$; dit noemen we een "beperkte kwantor".

Definitie 5.9 *Een formule die opgebouwd is uit de volgende bouwstenen alleen:*

i) *atomaire formules en negaties van atomaire formules*

ii) $\exists$, $\wedge$, $\vee$

iii) *beperkte kwantoren* $\forall i(\varphi_R(i, t) \rightarrow \cdots)$

noemen we een Σ_1-*formule.*

Propositie 5.10 *Elke primitief recursieve functie is nummersgewijs representeerbaar door een* Σ_1-*formule.*

Bewijs. De functies die nummersgewijs representeerbaar zijn door een Σ_1-formule, bevatten de basisfuncties en zijn, blijkens het bewijs van stelling 5.8, gesloten onder primitieve recursie; merk op dat in de definitie van φ_F de formules φ_G, φ_H en B "positief" voorkomen (d.w.z. rechts van de implicatiepijl). Ga zelf na dat ze ook gesloten zijn onder compositie. ■

5.4 Codering van de taal $\mathcal{L}_{PA}$ en het diagonalisatielemma

We beginnen met een "woordenboek" van tekens aan te leggen:

$\forall$	$\exists$	$\wedge$	$\vee$	$\rightarrow$	$\neg$	0	S	$+$	$\cdot$	v	$=$
1	2	3	4	5	6	7	8	9	10	11	12

Definitie 5.11 *We kennen aan elke $\mathcal{L}_{PA}$-term t een* code $\ulcorner t \urcorner \in \mathbb{N}$ *toe met recursie op t, als volgt. We nemen aan de variabelen van onze logica genummerd zijn als $v_0, v_1, \ldots$. Zet nu (zie het woordenboek):*

$$
\begin{array}{rcl}
\ulcorner v_i \urcorner & = & \langle 11, i\rangle \\
\ulcorner 0 \urcorner & = & \langle 7\rangle \\
\ulcorner S(t) \urcorner & = & \langle 8, \ulcorner t \urcorner\rangle \\
\ulcorner t+s \urcorner & = & \langle 9, \ulcorner t \urcorner, \ulcorner s \urcorner\rangle \\
\ulcorner t \cdot s \urcorner & = & \langle 10, \ulcorner t \urcorner, \ulcorner s \urcorner\rangle
\end{array}
$$

Opgave 95. Bewijs dat de relatie Term: Term(x) dan en slechts dan als x de code is van een $\mathcal{L}_{PA}$-term, primitief recursief is.

Definitie 5.12 *We kennen aan iedere $\mathcal{L}_{PA}$-formule φ een code $\ulcorner \varphi \urcorner \in \mathbb{N}$ toe, met recursie op φ, op de voor de hand liggende manier (zie weer het woordenboek):*

$$
\begin{array}{rcl}
\ulcorner t = s \urcorner & = & \langle 12, \ulcorner t \urcorner, \ulcorner s \urcorner\rangle \\
\ulcorner \varphi \wedge \psi \urcorner & = & \langle 3, \ulcorner \varphi \urcorner, \ulcorner \psi \urcorner\rangle \\
\vdots & & \vdots \\
\ulcorner \forall v_i \varphi \urcorner & = & \langle 1, i, \ulcorner \varphi \urcorner\rangle \\
\ulcorner \exists v_i \varphi \urcorner & = & \langle 2, i, \ulcorner \varphi \urcorner\rangle
\end{array}
$$

Er valt nu een hele waslijst van eigenschappen te controleren, die we voor het merendeel als opgaven stellen.

Opgave 96. Het predikaat Form(x): "x codeert een $\mathcal{L}_{PA}$-formule", is primitief recursief.

Opgave 97. Het predikaat Varvoork(y, x): "x codeert een formule φ, en de variabele v_y komt voor in φ", is primitief recursief.

Opgave 98. Evenzo: Vrijvoork(y, x): "$x = \ulcorner \varphi \urcorner$ en v_y komt vrij voor in φ", en VrijVoor(z, y, x): "$x = \ulcorner \varphi \urcorner$ en $z = \ulcorner s \urcorner$ en s is vrij voor v_y in φ".

In wezen gaan al deze opgaven op dezelfde manier. Bijvoorbeeld Form(x) geldt dan en slechts dan als

$$
\begin{array}{rl}
\exists y, z < x (x = \langle 12, y, z\rangle \wedge \mathrm{Term}(y) \wedge \mathrm{Term}(z)) & \vee \\
\exists y, z < x (x = \langle 3, y, z\rangle \wedge \mathrm{Form}(y) \wedge \mathrm{Form}(z)) & \vee \\
\vdots & \vdots
\end{array}
$$

Omdat onze rijtjescodering zo is dat $x, y, z < \langle x, y, z\rangle$ voor alle x, y, z geeft dit een definitie van de karakteristieke functie χ_{Form}, met recursie naar waardenverloop. Op deze manier kunnen we vrijwel alle recursief gedefinieerde begrippen uit de logica omzetten in primitief recursieve predikaten en operaties op de codes.

Nog een voorbeeld: de functie Tsub(x, i, y), die $\ulcorner s[t/v_i] \urcorner$ geeft als $y = \ulcorner s \urcorner$ en $x = \ulcorner t \urcorner$, en 0 anders.

$$
\mathrm{Tsub}(x, i, y) = \left\{
\begin{array}{rl}
0 & \text{als } \neg(\mathrm{Term}(x) \wedge \mathrm{Term}(y)) \\
y & \text{als } y = \langle 7\rangle \text{ of } y = \langle 11, j\rangle \text{ met } j \neq i \\
x & \text{als } y = \langle 11, i\rangle \\
\langle 8, \mathrm{Tsub}(x, i, z)\rangle & \text{als } y = \langle 8, z\rangle \\
\langle 9, \mathrm{Tsub}(x, i, z), \mathrm{Tsub}(x, i, w)\rangle & \text{als } y = \langle 9, z, w\rangle \\
\langle 10, \mathrm{Tsub}(x, i, z), \mathrm{Tsub}(x, i, w)\rangle & \text{als } y = \langle 10, z, w\rangle
\end{array}
\right.
$$

Opgave 99. Definieer zelf analoog een primitief recursieve functie Sub(x, i, y), die $\ulcorner \varphi[s/v_i] \urcorner$ geeft als $y = \ulcorner \varphi \urcorner$ en $x = \ulcorner s \urcorner$, en 0 anders.

De pointe van al deze definities (en bewijzen van primitief-recursiviteit) is natuurlijk dat al deze predikaten en functies nu, m.b.v. stelling 5.8, nummersgewijs representeerbaar zijn.

Conventie. We voeren nu de volgende afkortingsconventie in. Als $F : \mathbb{N}^k \to \mathbb{N}$ primitief recursief is en φ_F representeert F in PA dan schrijven we, als $t_1, \ldots, t_k$ $\mathcal{L}_{PA}$-termen zijn, $F(t_1, \ldots, t_k)$ alsof het een $\mathcal{L}_{PA}$-term is. Als dit in een formule voorkomt, zeg in $\psi[F(t_1, \ldots, t_k)/z]$ vatten we dit op als afkorting van $\exists w(\varphi_F(t_1, \ldots, t_k, w) \wedge \psi[w/z])$.

We zijn nu klaar voor de belangrijkste bouwsteen van het bewijs van Gödels eerste onvolledigheidsstelling.

Lemma 5.13 (Diagonalisatielemma; ook wel dekpuntsstelling genoemd) *Laat φ een $\mathcal{L}_{PA}$-formule zijn met vrije variabele v_0. Dan is er een $\mathcal{L}_{PA}$-formule ψ zodat $FV(\psi) = FV(\varphi) \setminus \{v_0\}$ (met $FV(\psi)$ geven we de collectie vrije variabelen van ψ aan), zodat*

$$PA \vdash \psi \leftrightarrow \varphi[\overline{\ulcorner\psi\urcorner}/v_0]$$

Bewijs. Laat C de formule $\varphi[\mathrm{Sub}(v_0, 0, v_0)/v_0]$ zijn (denk aan onze afkortingsconventie!) en laat

$$\psi \equiv \varphi[\mathrm{Sub}(\overline{\ulcorner C\urcorner}, 0, \overline{\ulcorner C\urcorner})/v_0]$$

Dan geldt:

$$PA \vdash \left\{ \begin{array}{rc} \mathrm{Sub}(\overline{\ulcorner C\urcorner}, 0, \overline{\ulcorner C\urcorner}) & = \\ \ulcorner C[\overline{\ulcorner C\urcorner}/v_0]\urcorner & = \\ \ulcorner \varphi[\mathrm{Sub}(\overline{\ulcorner C\urcorner}, 0, \overline{\ulcorner C\urcorner})/v_0]\urcorner & = \\ \ulcorner\psi\urcorner & = \\ \overline{\ulcorner\psi\urcorner} & \end{array} \right.$$

Er volgt:

$$PA \vdash \varphi[\overline{\ulcorner\psi\urcorner}/v_0] \leftrightarrow \varphi[\mathrm{Sub}(\overline{\ulcorner C\urcorner}, 0, \overline{\ulcorner C\urcorner})/v_0] \equiv \psi$$

■

Merk de analogie op tussen formulering en bewijs van stelling 5.13 en die van de recursiestelling! Historisch is echter het diagonalisatielemma ouder; in feite is Gödels onvolledigheidsstelling een belangrijke inspiratiebron geweest voor de ontwikkeling van de recursietheorie.

5.5 Codering van bewijzen en voltooiing van het bewijs van Gödels stelling

Volledig analoog aan de codering van de syntaxis in paragraaf 4 kunnen we nu ook bewijsbomen van de natuurlijke deductie coderen. Ik doe dit enigszins schetsmatig omdat er geen wezenlijk nieuwe ideeën aan ten grondslag liggen.

Leg weer net als in het begin van paragraaf 4, een woordenboek aan, nu van alle natuurlijke deductieregels. Een gedeelte:

$$\begin{array}{ccccc} \frac{\varphi}{\varphi} & \frac{}{x=x} & \wedge I : \frac{\varphi \;\; \psi}{\varphi\wedge\psi} & \forall I : \frac{\varphi}{\forall x\varphi} & \to I : \frac{\varphi}{\psi\to\varphi} \\ 1 & 2 & 3 & 4 & 5 \end{array}$$

De boom $\frac{\varphi}{\varphi}$ krijgt code $\langle 1, \ulcorner\varphi\urcorner\rangle$, $\frac{}{x=x}$ krijgt $\langle 2, \ulcorner x = x\urcorner\rangle$; als $\frac{D}{\varphi}$ en $\frac{D'}{\psi}$ bomen zijn met codes x en y krijgt $\dfrac{\frac{D}{\varphi} \;\; \frac{D'}{\psi}}{\varphi\wedge\psi}$ code $\langle 3, x, y, \ulcorner\varphi \wedge \psi\urcorner\rangle$; als $\frac{D}{\varphi}$ code x heeft krijgt $\dfrac{\frac{D}{\varphi}}{\forall v_i\varphi}$ code $\langle 4, i, x, \ulcorner\forall v_i\varphi\urcorner\rangle$.

Zo is iedere code van een bewijsboom een rijtje waarvan het *eerste* element de *laatst toegepaste regel* codeert, en het laatste element de *conclusie*.

Nu zijn niet alle op deze manier geconstrueerde codes van bomen ook codes van *bewijs*bomen; bijvoorbeeld toepassing van de regel $\forall I$ is onderworpen aan de restrictie dat de variabele niet vrij mag voorkomen in enige hypothese van de boom.

We definiëren dus een primitief recursief predicaat $\mathrm{Hyp}(x, y)$: x codeert formule φ en y een formuleboom B en φ komt voor als hypothese in B. Dit gaat met een evidente recursie op de formuleboom B. Een klein stukje:

$$\mathrm{Hyp}(x, y) \Leftrightarrow \begin{cases} \begin{array}{rr} [y = \langle 1, x\rangle & \vee \\ \exists z, w, v < y(y = \langle 3, z, w, v\rangle \wedge (\mathrm{Hyp}(x, z) \vee \mathrm{Hyp}(x, w))) & \vee \\ \vdots &] \\ \wedge & \\ y \neq \langle 2, \ulcorner v_i = v_i \urcorner\rangle & \wedge \\ y \neq \langle 5, z, \langle 5, x, w\rangle\rangle & \wedge \\ \vdots & \end{array} \end{cases}$$

De laatste regel zegt, dat y niet een boom codeert met als laatste stap een $\rightarrow$-introductie waarbij juist de formule gecodeerd door x wordt doorgestreept als hypothese (het is gerieflijk om hier te werken met een natuurlijk deductiesysteem waarin, boven elke $\rightarrow$-introductie $\varphi \rightarrow \psi$, *elk* voorkomen van φ wordt doorgestreept als hypothese). Er is natuurlijk ook nog een ingewikkelde voorwaarde bij een regel als $\exists$-eliminatie. . .

Vervolgens construeren we een primitief recursief predikaat $\mathrm{Bewijs}(x)$: x codeert een formuleboom, en bij elke intrductie van $\forall v_i \varphi$ komt v_i niet voor in enige hypothese van de relevante subboom. Er zijn natuurlijk meer voorwaarden, maar dit laat ik aan de fantasie van delezer over.

We hebben ook een primitief recursief predikaat $\mathrm{Ax}(x)$: x codeert een axioma van PA (ga na dat de codes van de axioma's een primitief recursieve verzameling vormen: alle inductie-axioma's zijn van buiten van dezelfde vorm).

Opgave 100. Schrijf dit uit, d.w.z. definieer $\mathrm{Ax}(x)$ als primitief recursief predikaat.

Tenslotte is er een predikaat Bew_{PA}: $\mathrm{Bew}_{PA}(x, y)$ geldt precies als x een bewijsboom codeert met conclusie de formule gecodeerd door y, en alle hypothesen van de boom zijn PA-axioma's.

Ook dit predikaat is primitief recursief en dus nummersgewijs representeerbaar in PA. Laat $\overline{\mathrm{Bew}}(x, y)$ de $\mathcal{L}_{PA}$-formule zijn die Bew_{PA} representeert.

Lemma 5.14

i) $PA \vdash \varphi \Rightarrow PA \vdash \exists x \overline{Bew}(x, \overline{\ulcorner \varphi \urcorner})$

ii) $PA \vdash \exists x \overline{Bew}(x, \overline{\ulcorner \varphi \rightarrow \psi \urcorner}) \wedge \exists x \overline{Bew}(x, \overline{\ulcorner \varphi \urcorner}) \rightarrow \exists x \overline{Bew}(x, \overline{\ulcorner \psi \urcorner})$

Bewijs. i) Als $PA \vdash \varphi$ is er een bewijsboom voor φ waarvan alle hypothesen PA-axioma's zijn. Voor zekere n (nl. de code van zo'n boom) geldt dan $\mathrm{Bew}_{PA}(n, \ulcorner \varphi \urcorner)$. Omdat $\overline{\mathrm{Bew}}$ Bew_{PA} representeert volgt dat $PA \vdash \overline{\mathrm{Bew}}(\overline{n}, \overline{\ulcorner \varphi \urcorner})$ zodat $PA \vdash \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner})$.

ii) Stel de code in ons woordenboek voor de regel

$$\rightarrow E \quad \frac{\varphi \rightarrow \psi \quad \varphi}{\psi}$$

is k. Als (in PA!) $\overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \rightarrow \psi \urcorner})$ en $\overline{\mathrm{Bew}}(y, \overline{\ulcorner \varphi \urcorner})$ dan volgt $\overline{\mathrm{Bew}}(\langle \overline{k}, x, y, \overline{\ulcorner \psi \urcorner}\rangle, \overline{\ulcorner \psi \urcorner})$. ■

Opgave 101. Is het wel zo evident wat in het bewijs van lemma 5.14ii) beweerd wordt? Schets een exact bewijs.

Stelling 5.15 (Gödels eerste onvolledigheidsstelling) *PA is onvolledig.*

Bewijs. Pas het diagonalisatielemma (5.13) toe op de formule $\neg \exists x \overline{\mathrm{Bew}}(x, v_0)$. Er is dus een $\mathcal{L}_{PA}$-zin ψ zodat

$$PA \vdash \psi \leftrightarrow \neg \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \psi \urcorner})$$

Dan geldt: $PA \not\vdash \psi$ en $PA \not\vdash \neg\psi$.

Immers, stel $PA \vdash \psi$. Per definitie van ψ volgt $PA \vdash \neg\exists x\overline{\text{Bew}}(x, \overline{\ulcorner\psi\urcorner})$. Uit lemma 5.14(i) volgt echter ook $PA \vdash \exists x\overline{\text{Bew}}(x, \overline{\ulcorner\psi\urcorner})$. Er volgt: PA is inconsistent, quod non. Want PA heeft een model, zoals we gezien hebben.

Stel $PA \vdash \neg\psi$; per definitie volgt $PA \vdash \exists z\overline{\text{Bew}}(x, \overline{\ulcorner\psi\urcorner})$. Uit lemma 5.14i) volgt ook $PA \vdash \exists x\overline{\text{Bew}}(x, \overline{\ulcorner\neg\psi\urcorner})$. Door tweemaal toepassen van lemma 5.14 volgt dan $PA \vdash \exists x\overline{\text{Bew}}(x, \overline{\ulcorner\bot\urcorner})$. Omdat $\mathcal{N}$ een model van PA is geldt dus voor zekere $n \in \mathbb{N}$: $\mathcal{N} \models \overline{\text{Bew}}(\overline{n}, \overline{\ulcorner\bot\urcorner})$. Omdat $\overline{\text{Bew}}$ Bew_{PA} representeert, volgt dat $PA \vdash \bot$, en weer een tegenspraak. ∎

De zin ψ geconstrueerd in het bewijs van stelling 5.15 is de befaamde "Gödel-zin". Hij heeft velen in extase gebracht.

De zin die, in PA, uitdrukt dat PA consistent is, is de zin

$$\neg\exists x\overline{\text{Bew}}(x, \overline{\ulcorner\bot\urcorner})$$

Deze zin wordt vaak aangeduid met Con_{PA}. In de volgende paragraaf zullen we zien dat

$$PA \vdash \psi \leftrightarrow \text{Con}_{PA}$$

zodat uit 5.15 volgt dat $PA \not\vdash \text{Con}_{PA}$. Oftewel: "$PA$ bewijst zijn eigen consistentie niet". Dit feit is Gödels tweede onvolledigheidsstelling.

We zullen ook zien dat dit geen speciale eigenschap van PA is: elke $\mathcal{L}_{PA}$-theorie die PA uitbreidt met een recursief opsombare verzameling axioma's, is onvolledig op deze manier. Evenmin speelt $\mathcal{L}_{PA}$ een speciale rol: voldoende is een theorie in een taal waarin we over getallen kunnen praten en waarin primitief recursieve functies (niet eens alle!) representeerbaar zijn; de theorie moet wel een recursief opsombare verzameling axioma's hebben. Dit gaat bijvoorbeeld op voor Zermelo-Fraenkel verzamelingenleer ZF.

5.6 Σ_1-volledigheid en Gödels tweede onvolledigheidsstelling

In deze paragraaf is de hoofdzaak het bewijs van Gödels tweede onvolledigheidsstelling. Hierbij komt een kwestie aan de orde die ook voor het bewijs van de stelling van Church van belang zal zijn, nl. de Σ_1-volledigheid van PA. Dit is de bewering dat PA alle ware Σ_1-zinnen bewijst.

Definitie 5.16 *De verzameling Δ_0-formules van $\mathcal{L}_{PA}$ is opgebouwd uit:*

1. *alle atomaire formules;*
2. *alle propositielogische tekens;*
3. *de "beperkte" kwantoren $\forall x < t$ en $\exists x < t$ (waar t een term is die x niet als vrije variabele bevat)*

Opgave 102. Bewijs m.b.v. de inductieaxioma's van PA het zg. *collectieprincipe*:

$$PA \vdash \forall y < t\exists x A(x, y) \rightarrow \exists x\forall y < t\exists u < x A(u, y)$$

Opgave 103. Bewijs m.b.v. de vorige opgave: voor elke Σ_1-formule φ is er een Δ_0-formule ψ en een (mogelijk leeg) rijtje variabelen $\vec{x}$ zodat

$$PA \vdash \varphi \leftrightarrow \exists\vec{x}\psi$$

Propositie 5.17 *Voor elke Δ_0-formule φ met vrije variabelen $x_1, \ldots, x_n$ en voor alle $m_1, \ldots, m_n \in \mathbb{N}$:*

$$\mathcal{N} \models \varphi[m_1, \ldots, m_n] \Leftrightarrow PA \vdash \varphi[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n]$$

Bewijs. Inductie op φ. Merk op dat $m < n \Leftrightarrow PA \vdash \overline{m} < \overline{n}$, $m + n = k \Leftrightarrow PA \vdash \overline{m} + \overline{n} = \overline{k}$, etc.

De inductiestap voor propositionele connectieven is triviaal. Als nu $\varphi \equiv \forall y < t\psi$ met $y \notin \mathrm{FV}(t)$, bewijs eerst de bewering dat voor alle $m_1, \ldots, m_{n+1}$:

$$\mathcal{N} \models \forall y < m_{n+1}\psi[m_1, \ldots, m_n] \Leftrightarrow PA \vdash \forall y < \overline{m_{n+1}}\psi[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n]$$

met inductie naar m_{n+1}; gebruik hierbij dat $PA \vdash \forall y(y < S(z) \leftrightarrow y = z \vee y < z)$. Merk dan op, dat

$$(t[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n])^{\mathcal{N}} = k \Rightarrow PA \vdash t[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n] = \overline{k}$$

Dus

$$\begin{array}{rl} \mathcal{N} \models \forall y < t\psi[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n] & \Rightarrow \\ \mathcal{N} \models \forall y < \overline{k}\psi[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n] & \Rightarrow \\ PA \vdash \forall y < \overline{k}[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n] & \Rightarrow \\ PA \vdash \forall y < t\psi[\overline{m_1}/x_1, \ldots, \overline{m_n}/x_n] & \end{array}$$

De inductiestap voor $\exists y < t$ is duaal en wordt aan de lezer overgelaten. ∎

Gevolg 5.18 (Σ_1-volledigheid van PA) *Voor Σ_1-zinnen $\varphi \in \mathcal{L}_{PA}$ geldt:*

$$\mathcal{N} \models \varphi \Leftrightarrow PA \vdash \varphi$$

Bewijs. Want als $PA \vdash \varphi \leftrightarrow \exists x_1, \ldots, x_k\psi$ met $\psi \in \Delta_0$ dan geldt

$$\begin{array}{lrll} & \mathcal{N} \models \varphi & \Rightarrow & \\ \text{voor zekere } n_1, \ldots, n_k \in \mathbb{N} & \mathcal{N} \models \psi[n_1, \ldots, n_k] & \Rightarrow & \text{met 5.17} \\ \text{voor zekere } n_1, \ldots, n_k \in \mathbb{N} & PA \vdash \psi[\overline{n_1}/x_1, \ldots, \overline{n_k}/x_k] & \Rightarrow & \\ & PA \vdash \exists x_1, \ldots, x_k\psi & & \end{array}$$

∎

Nu is er, geheel analoog met de functie $\mathrm{Sub}(x, i, y)$ een functie $\mathrm{Sub}'(x_1, \ldots, x_n, i_1, \ldots, i_n, y)$ zodat

$$\mathrm{Sub}'(\ulcorner s_1 \urcorner, \ldots, \ulcorner s_n \urcorner, i_1, \ldots, i_n, \ulcorner \varphi \urcorner) = \ulcorner \varphi[s_1/v_{i_1}, \ldots, s_n/v_{i_n}] \urcorner$$

Sub' is primitief recursief.

Propositie 5.17 kan nu zó worden gelezen: voor $\varphi \in \Delta_0$ met vrije variabelen $v_{i_1}, \ldots, v_{i_n}$, en $m_1, \ldots, m_n \in \mathbb{N}$:

$$\mathcal{N} \models \varphi[m_1, \ldots, m_n] \Rightarrow \exists k \mathrm{Bew}_{PA}(k, \mathrm{Sub}'(\ulcorner \overline{m_1} \urcorner, \ldots, \ulcorner \overline{m_n} \urcorner, i_1, \ldots, i_n, \ulcorner \varphi \urcorner))$$

De juiste "formalisering" van deze bewering in PA luidt nu:

Propositie 5.19 *Voor $\varphi \in \Delta_0$ met vrije variabelen $v_{i_1}, \ldots, v_{i_n}$ geldt:*

$$PA \vdash \forall x_1 \cdots x_n(\varphi[x_1/v_{i_1}, \ldots, x_n/v_{i_n}] \to \exists k\overline{Bew}(k, F(x_1, \ldots, x_n)))$$

waar F de primitief recursieve functie

$$\lambda m_1 \cdots m_n.Sub'(\ulcorner \overline{m_1} \urcorner, \ldots, \ulcorner \overline{m_n} \urcorner, i_1, \ldots, i_n, \ulcorner \varphi \urcorner)$$

is.

Bewijs. Dit gaat met inductie naar φ en inductie (in PA) naar $x_1, \ldots, x_n$, in volledige analogie met propositie 5.17. We schrijven dit hier niet uit. ∎

Gevolg 5.20 (geformaliseerde Σ_1-volledigheid van PA) *Voor Σ_1-zinnen φ geldt*

$$PA \vdash \varphi \to \exists x\overline{Bew}(x, \overline{\ulcorner \varphi \urcorner})$$

Bewijs. Laat $PA \vdash \varphi \leftrightarrow \exists x_1 \cdots x_n \psi$ met $\psi \in \Delta_0$. Dan geldt wegens propositie 5.19

$$PA \vdash \forall x_1 \cdots x_n (\psi[x_1/v_{i_1}, \ldots, x_n/v_{i_n}] \rightarrow \exists k \overline{\text{Bew}}(k, F(x_1, \ldots, x_n)))$$

dus

$$PA \vdash \exists x_1 \cdots x_n \psi[x_1/v_{i_1}, \ldots, x_n/v_{i_n}] \rightarrow \exists k \overline{\text{Bew}}(k, \overline{\ulcorner \exists x_1 \cdots x_n \psi[x_1/v_{i_1}, \ldots, x_n/v_{i_n}] \urcorner})$$

oftewel $PA \vdash \varphi \rightarrow \exists x \overline{\text{Bew}}(x, \overline{\ulcorner \varphi \urcorner})$. ∎

Gevolg 5.21 *Laat ψ de Gödelzin uit 5.15 zijn. Dan:*

$$PA \vdash \psi \leftrightarrow \text{Con}_{PA}$$

Bewijs. Daar met 5.14ii) makkelijk volgt $PA \vdash \exists x \overline{\text{Bew}}(x, \overline{\ulcorner \bot \urcorner}) \rightarrow \exists x \overline{\text{Bew}}(x, \overline{\ulcorner \psi \urcorner})$, is $PA \vdash \psi \rightarrow \text{Con}_{PA}$ duidelijk. Merk op dat $\neg\psi$ equivalent is met een Σ_1-zin, zodat we 5.20 kunnen toepassen:

$$PA \vdash \neg\psi \rightarrow \exists x \overline{\text{Bew}}(x, \overline{\ulcorner \neg\psi \urcorner})$$

Per definitie van ψ geldt ook

$$PA \vdash \neg\psi \rightarrow \exists x \overline{\text{Bew}}(x, \overline{\ulcorner \psi \urcorner})$$

zodat met 5.14 volgt

$$PA \vdash \neg\psi \rightarrow \exists x \overline{\text{Bew}}(x, \overline{\ulcorner \bot \urcorner})$$

en met contrapositie

$$PA \vdash \text{Con}_{PA} \rightarrow \psi$$

∎

Gevolg 5.22 (Gödels tweede onvolledigheidsstelling) $PA \nvdash \text{Con}_{PA}$

Bewijs. Onmiddellijk. ∎

Overzien we nu wat we gebruikt hebben voor het bewijs, dan is dit:

- Primitief recursieve functies en relaties zijn nummersgewijs representeerbaar in PA;
- PA heeft een primitief recursieve verzameling axioma's;
- PA is consistent, en bewijst alleen *ware* zinnen: $PA \vdash \varphi \Rightarrow \mathcal{N} \models \varphi$

In feite gebruiken we de laatste eigenschap niet helemaal: het is voldoende te weten dat PA alleen ware Σ_1-zinnen bewijst (dit feit noemt men de *ω-consistentie* van PA).

Met kleine verfijningen van het bewijs kunnen deze voorwaarden nog aanzienlijk worden afgezwakt. Er geldt: als T een theorie is in de predikatenlogica zodat

- Primitief recursieve functies en relaties zijn nummersgewijs representeerbaar in T
- T heeft een recursief opsombare verzameling axioma's
- T is consistent

dan geldt $T \nvdash \text{Con}_T$.

Dat de tweede voorwaarde afgezwakt kon worden, is wel te begrijpen. Er is geen primitief recursief predikaat $\text{Ax}(x)$ meer, maar het predikaat $\text{Bew}_T(x, y)$: "x codeert een bewijs in T van de formule gecodeerd door y", blijft recursief opsombaar en dus (op grond van de eerste voorwaarde) definieerbaar door een Σ_1-formule.

Dat de voorwaarde: $T \vdash \varphi \Rightarrow \mathcal{N} \models \varphi$ kan vervallen (zelfs voor $\varphi \in \Sigma_1$), werd enkele jaren na Gödels resultaat aangetoond door Rosser. Deze versterking is niet van belang ontbloot, want zij geeft ons zeer veel modellen van PA:

Wegens $PA \nvdash \mathrm{Con}_{PA}$ is $PA + \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \bot \urcorner})$ consistent; noem deze theorie PA_1. Deze heeft dus een model, waarin zich een getal bevindt dat een "PA-bewijs van $\bot$ codeert". Passen we de Gödelconstructie op PA_1 toe, dan vinden we $PA_1 \nvdash \mathrm{Con}_{PA_1}$ en dus is $PA_2 := PA_1 + \exists x \overline{\mathrm{Bew}}_1(x, \overline{\ulcorner \bot \urcorner})$ consistent ($\overline{\mathrm{Bew}}_1$ representeert dan Bew_{PA_1}); etcetera.

Een ander toepassinkje van dit idee is de *Stelling van Löb*. We hebben een paar keer de redenering gezien: "$PA \vdash \varphi$ dus φ is waar in $\mathcal{N}$", oftewel de zin $\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner}) \to \varphi$ is waar in $\mathcal{N}$. Vraag: voor welke φ is $\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner}) \to \varphi$ *bewijsbaar* in PA?

Opgave 104. Laat zien dat $PA \nvdash \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \psi \urcorner}) \to \psi$ als ψ de Gödelzin is.

De stelling van Löb luidt, dat het alleen maar op een triviale manier kan, nl. als φ zèlf al bewijsbaar is in PA:

Stelling 5.23 (stelling van Löb) *Als $PA \vdash \exists x \overline{Bew}(x, \overline{\ulcorner \varphi \urcorner}) \to \varphi$, dan $PA \vdash \varphi$.*

Bewijs. Stel $PA \vdash \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner}) \to \varphi$, $PA \nvdash \varphi$. Dan is $PA + \neg\varphi$ consistent en kunnen we de Gödelconstructie op $PA + \neg\varphi$ toepassen. Er volgt:

$$PA + \neg\varphi \nvdash \mathrm{Con}_{PA+\neg\varphi}$$

Nu geldt uiteraard

$$PA \vdash \mathrm{Con}_{PA+\neg\varphi} \leftrightarrow \neg\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \neg\varphi \to \bot \urcorner})$$

Dus $PA + \neg\varphi \nvdash \neg\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \neg\varphi \to \bot \urcorner})$. Maar dan volgt ook $PA + \neg\varphi \nvdash \neg\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner})$ met 5.14i), omdat natuurlijk $\vdash \varphi \leftrightarrow (\neg\varphi \to \bot)$.

Dus $PA \nvdash \neg\varphi \to \neg\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner})$ zodat met contrapositie $PA \nvdash \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \varphi \urcorner}) \to \varphi$, in strijd met de aanname. ∎

Opgave 105. Bewijs de stelling van Löb direkt vanuit het diagonalisatielemma:

1. door een ψ te nemen zodat

$$PA \vdash \psi \leftrightarrow \exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \psi \to \varphi \urcorner})$$

2. door een ψ te nemen zodat

$$PA \vdash \psi \leftrightarrow (\exists x \overline{\mathrm{Bew}}(x, \overline{\ulcorner \psi \urcorner}) \to \varphi)$$

Gebruik propositie 5.14, Σ_1-volledigheid en geformaliseerde Σ_1-volledigheid van PA.

Opgave 106. **(Paradox van de gehangene)** Op een zaterdag zegt de rechter tegen de veroordeelde: "Tussen nu en volgende week zult gij op een dag om 10 uur 's morgens gehaald en gehangen worden; en gij zult de dag van tevoren niet weten."

De man redeneert nu als volgt. Het laatste tijdstip waarop ik gehangen kan worden is dus volgende week zaterdag om 10 uur. Maar dan kan ik niet gehangen worden, want ik zou het een dag lang van tevoren weten. Dus op z'n laatst vrijdag om 10 uur. Maar dan wist ik het al vanaf donderdag om vijf over tien... Doorredenerend, komt hij tot de conclusie dat hij niet gehangen kan worden, en slaapt opgelucht in.

Hij wordt op donderdag om 10 uur gehaald en gehangen; en hij wist het niet van tevoren.

1. Had hij ook op zaterdag nog gehaald kunnen worden?

2. Verklaar de paradox.

5.7 Stellingen van Church en Tarski

Tenslotte behandelen we nog twee veel eenvoudiger stellingen uit de logica, die ook met codering van formules te maken hebben.

De *Stelling van Church* luidt, dat de predikatenlogica onbeslisbaar is, d.w.z. dat er geen algoritme is dat voor iedere formule φ in de zuivere predikatenlogica beslist of φ bewijsbaar is of niet. Ik bewijs: er is een eindig geaxiomatiseerde theorie T in een eindige taal $\mathcal{L}$ in de predikatenlogica, zodat de verzameling $\{\ulcorner\varphi\urcorner \,|\, T \vdash \varphi\}$ niet recursief is.

Dit wil natuurlijk niet zeggen dat èlke eindig geaxiomatiseerde theorie onbeslisbaar is: het kost niet veel moeite om in te zien, dat elke *volledige* theorie die recursief opsombaar is, ook beslisbaar is.

Laat $\mathcal{L} = \mathcal{L}_{PA}$. Om één primitief recursief predikaat R nummersgewijs in PA te representeren heb je maar eindig veel inductieaxioma's nodig, evenals om te laten zien dat voor een primitief recursief predikaat $R(x_1, \ldots, x_{n+1})$:

$$\exists m_{n+1} R(m_1, \ldots, m_{n+1}) \Rightarrow PA \vdash \exists y \overline{R}(\overline{m_1}, \ldots, \overline{m_n}, y)$$

(waar $\overline{R}$ R representeert; dit noem ik de Σ_1-volledigheid voor R)

Laat nu S de theorie zijn geaxiomatiseerd door: de axioma's (1)–(4) van PA, plus precies die inductieaxioma's, nodig om Kleene's T-predikaat $T(1, e, x, y)$ nummersgewijs te representeren en de Σ_1-volledigheid hiervoor te bewijzen.

Dan is $\{\ulcorner\varphi\urcorner \mid S \vdash \varphi\}$ niet recursief; immers stel $\overline{T}$ representeert T, dan geldt $x \bullet x$ gedefinieerd $\Leftrightarrow$ $S \vdash \exists y \overline{T}(\overline{1}, \overline{x}, \overline{x}, y)$ en omdat de code $\ulcorner \exists y \overline{T}(\overline{1}, \overline{x}, \overline{x}, y)\urcorner$ een primitief recursieve functie van x is, kunnen we het standaardprobleem $\mathcal{K}$ reduceren tot het probleem "$S \vdash \varphi$?".

Merk op dat dezelfde redenering aantoont dat $\{\ulcorner\varphi\urcorner \mid PA \vdash \varphi\}$ ook strict r.e. is. Vraag: hoe complex is de verzameling $\{\ulcorner\varphi\urcorner \mid \mathcal{N} \models \varphi\}$?

Het antwoord hierop wordt gegeven door de stelling van Tarski, die uitspreekt dat deze verzameling niet arithmetisch is:

Stelling 5.24 (Tarski) *Er is geen $\mathcal{L}_{PA}$-formule met één vrije variabele v_0, zodat voor alle $\mathcal{L}_{PA}$-zinnen ψ geldt:*

$$\mathcal{N} \models \psi \Leftrightarrow \mathcal{N} \models \varphi[\ulcorner\psi\urcorner]$$

Bewijs. Dit is een diagonalisatie zoals we er al meer gezien hebben. Stel φ als in de stelling. Er is een primitief recursieve opsomming $\{\varphi_k \mid k \in \mathbb{N}\}$ van alle $\mathcal{L}_{PA}$-formules met precies één vrije variabele v_0; dan is de functie

$$F(k, n) = \ulcorner \varphi_k[\overline{n}/v_0]\urcorner$$

primitief recursief.

Zij nu $\zeta(v_0)$ de formule $\neg\varphi[F(v_0, v_0)/v_0]$. Dan komt ζ in onze opsomming voor, zeg als φ_{k_0}. Er geldt echter:

$$\begin{array}{ll}
\mathcal{N} \models \zeta[k_0] & \Leftrightarrow \\
\mathcal{N} \models \zeta[\overline{k_0}/v_0] & \Leftrightarrow \\
\mathcal{N} \models \varphi[\ulcorner\zeta[\overline{k_0}/v_0]\urcorner] & \Leftrightarrow \\
\mathcal{N} \models \varphi[F(k_0, k_0)] & \Leftrightarrow \\
\mathcal{N} \models \neg\zeta[k_0] &
\end{array}$$

Een overduidelijke paradox is bereikt. ∎

Literatuur. Gödels oorspronkelijke artikel is *Über formal unentscheidbare Sätze der Principia Mathematica und verwandte Systeme I*, Monatshefte für Mathematik und Physik **38**, pp. 173-198 (1931). De tweede onvolledigheidsstelling, daar aangekondigd, werd voor het eerst geheel bewezen in:

Hilbert en Bernays, *Grundlagen der Mathematik I* (1934) en *II* (1939). Latere edities in 1968 en 1970.

Er zijn tal van moderne exposities van Gödels onvolledigheidsstellingen. Schetsmatig in Smoryński's hoofdstuk in het *Handbook of Mathematical Logic* (ed. Barwise, North Holland 1977), uitvoeriger in o.a.:

S.C. Kleene, *Introduction to Metamathematics*
G. Boolos, *The Unprovability of Consistency*
C. Smoryński, *Self-reference and Modal Logic*
Shoenfield, *Mathematical Logic*
R. Smullyan, *Gödel's Incompleteness Theorems*
R. Kaye, *Models of Peano Arithmetic*

Index