Complexiteitsklassen in eerste-orde rekenkunde

Auteur: Caroline Kok
Supervisor: Jaap van Oosten
Bachelorprogramma: Wiskunde & Toepassingen
7,5 ECTS

16 juni 2017

Inhoudsopgave

Samenvatting

In deze scriptie zullen bijzondere deelsystemen van de Peano rekenkunde worden behandeld. De deelsystemen S_2^1 en T_2^1 zullen geïntroduceerd worden en daarbij worden hun sterkte en zogeheten bewijsbaar totale functies in relatie tot de polynomiale hiërarchie behandeld. De uiteindelijke opvallende conclusie die getrokken wordt is dat de functies die op een bepaalde manier representeerbaar zijn in deze twee deelsystemen, precies corresponderen met de functies en predicaten die berekenbaar zijn in polynomiale tijd. Dit wordt gezien als de tijd die uitvoerbaar of *feasible* wordt genoemd, dus deze functies zijn berekenbaar door de computers van nu.

1 Inleiding

In de computerwetenschap is een fundamentele vraag wat een efficiënte methode is om een gegeven functie te berekenen. Vanuit de computerwetenschap wordt naar deze vraag gekeken met behulp van Turing machines, maar ook vanuit de wiskundige logica zijn er veel studies gedaan naar dit onderwerp.
Er worden veelal axioma's gebruikt om een systeem te beschrijven, dit soort systemen wordt ook wel een deductief systeem genoemd. In de wiskundige logica heten deze systemen theorieën.
In het bijzonder zijn er theorieën in de zogeheten eerste-orde rekenkunde, die functies met bepaalde eigenschappen kunnen representeren.
Een functie f heet *berekenbaar* als er een Turing machine M is, zo dat M de functie kan berekenen. Als een functie f berekenbaar is, kan het zijn dat de rekentijd onbruikbaar lang is. Daarom wordt de eis gesteld dat de functies *feasibly computable* zijn, dus ze zijn berekenbaar door de computers van nu. In het algemeen worden de functies die uitvoerbaar berekenbaar zijn, de functies die berekenbaar zijn in polynomiale tijd genomen. Dus de tijd die het kost om de functie te berekenen is begrensd door een polynoom in de lengte van de input.

Het merendeel van deze scriptie behandelt de *begrensde rekenkunde*, dit zijn zwakkere fragmenten van een eerste-orde rekenkundig systeem: de Peano rekenkunde (*Peano Arithmetic*, of *PA*). De Peano rekenkunde bestaat uit een taal met slechts vier elementen en een zestal axioma's. Het mooie is dat met deze taal alle natuurlijke getallen aangeduid kunnen worden. Als er een isomorfisme te bedenken is tussen een bepaald probleem en de natuurlijke getallen, kan dit probleem worden uitgedrukt in de Peano rekenkunde.
Het is een feit dat alle berekenbare functies representeerbaar zijn in PA. In deze scriptie zullen verschillende deelsystemen van PA behandeld worden. Voor deze systemen geldt dat een functie f representeerbaar is in de theorie dan en slechts dan als f van een bepaalde complexiteitsklasse is.

De vraag waar naar gekeken wordt is: Gegeven een theorie R, welke functies kan R representeren? Of welke functiesymbolen mogen geïntroduceerd worden in R?
Als een functie representeerbaar is in een theorie, mag hij toegevoegd worden aan de taal van deze theorie en vrij gebruikt worden in inductie-axioma's.
De functies waar naar gezocht wordt zijn de *bewijsbaar totale* functies. Deze functies zijn belangrijk omdat, als ze bewijsbaar totaal zijn voor standaard modellen van een theorie, ze gedefinieerd zijn in alle, ook niet-standaard, modellen van een theorie.

Om een idee te krijgen van de verschillende complexiteitsklassen in de rekenkunde wordt in hoofdstuk 2 de polynomiale hiërarchie toegelicht. Hiernaast worden nog twee rekenkundige hiërarchieën gesteld, met begrensde kwantoren en onbegrensde kwantoren.
In hoofdstuk 3 wordt een aantal fragmenten van de eerste-orde rekenkunde toegelicht. Er wordt kennis gemaakt met de verschillende axioma's en de belangrijke definitie van *bewijsbaar totaal* ofwel *bewijsbaar recursieve* functies en predicaten in de eerste-orde rekenkunde wordt genoemd. Dit zijn de functies die gezocht werden in de hoofdvraag.
Daarna komt de begrensde rekenkunde aan bod in hoofdstuk 4. Hier wordt de basis gelegd voor de belangrijkste theorieën die in deze scriptie behandeld worden, namelijk S_2^i en T_2^i.
In hoofdstuk 5 wordt alle functies die bewijsbaar totaal zijn in die theorieën genoemd. Daarnaast wordt ook de relatie tussen S_2^i en T_2^i toegelicht.
In het laatste en ook belangrijkste hoofdstuk, hoofdstuk 6, komen de bevindingen uit de eerdere hoofdstukken bij elkaar en worden concluderende stellingen en gevolgen genoemd. Deze worden aan het einde samengevat in hoofdstuk 7.

2 Polynomiale tijd functies

In dit hoofdstuk worden verschillende definities behandeld, met als belangrijkste bevinding dat de functies berekenbaar in polynomiale tijd corresponderen met een hiërarchie van functies en predicaten in de begrensde rekenkunde.
Over het algemeen worden de definities gebruikt zoals Buss [4] ze in zijn proefschrift heeft gedefinieerd. Om een vollediger beeld te kunnen geven van de polynomiale hiërarchie worden verschillende definities uit [3] en [4] samengevoegd.
Een belangrijke verzameling van functies bestaat uit de functies die berekenbaar zijn in polynomiale tijd. Polynomiale tijd betekent dat het aantal stappen in het berekenen van de functie begrensd is door een polynoom van de lengte van de input. Deze functies zijn belangrijk omdat ze relatief snel berekend kunnen worden, en dit wordt als *feasible computation* beschouwd.
Als eerste wordt de volgende verzameling van functies een basis, om met behulp van compositie van functies uit deze verzameling functies berekenbaar in polynomiale tijd samen te stellen.

Definitie 1. [4] De verzameling B van functies van $\mathbb{N}^k$ naar $\mathbb{N}$ bestaat uit de volgende in polynomiale tijd berekenbare functies:

1. de nulfunctie 0;
2. de successorfunctie $x \mapsto Sx = x + 1$;
3. de *shift-right* functie $x \mapsto \lfloor \frac{1}{2}x \rfloor$;
4. de *shift-left* functie $x \mapsto 2 \cdot x$;
5. $(x, y) \mapsto x \leq y = \begin{cases} 1 & \text{als } x \leq y \\ 0 & \text{als } x > y \end{cases}$;
6. $(x, y, z) \mapsto Choice(x, y, z) = \begin{cases} y & \text{als } x > 0 \\ z & \text{als } x = 0 \end{cases}$.

Met deze functies kunnen nieuwe functies gevormd worden met behulp van compositie.
Naast deze functies zijn er ook definities nodig om eindige rijtjes die getallen representeren te definiëren. De rijtjes worden gedefinieerd met behulp van *Gödelnummers*. Het Gödelnummer voor de rij $a_1, \cdots, a_k$ wordt als volgt geconstrueerd. Schrijf alle a_i in binaire notatie om een rij van nullen, enen en komma's vormen. Schrijf deze rij in omgekeerde volgorde op en vervang elke 0 door 10, elke 1 door 11 en elke komma door 00. Dit Gödelnummer wordt genoteerd met $< a_1, \cdots, a_k >$. Een klein voorbeeld: het Gödelnummer van de rij 3,4,5 wordt als volgt gevonden. 3,4,5 wordt 11,100,101. Omdraaien geeft 101,001,11. Vervangen geeft (11101100101011001111) of 969.423.

Definitie 2. [4] De verzameling B^+ bestaat uit de functies uit B samen met de volgende functies

1. $\beta(i, < a_1, \cdots, a_n >) = \begin{cases} n & \text{als } i = 0 \\ a_i & \text{als } 0 < i \leq n \end{cases}$
   Deze functie heet Gödels $\beta-$functie.
2. $Truncate(< a_1, a_2, \cdots, a_n >) =< a_2, \cdots, a_n >$.
3. $a_0 * < a_1, \cdots, a_n >=< a_0, a_1, \cdots, a_n >$

Nu volgt een aantal basis definities gedefinieerd door Buss [4], die nodig zijn verder in dit hoofdstuk.

Definitie 3. [4] De lengte van de binaire representatie van x wordt genoteerd als $|x|$, ofwel $|x| = \lceil log_2(x + 1) \rceil$. Er geldt dat $|0| = 0$. Als x een vector $(x_1, \cdots, x_n)$ is, dan is $|x|$ de vector $(|x_1|, \cdots, |x_n|)$.

Definitie 4. Een functie f heeft polynomiale groei dan en slechts dan als er een polynoom p is zodat voor alle x geldt $|f(x)| \leq p(|x|)$.

Definitie 5. P is de verzameling van functies berekenbaar door een Turing machine in polynomiale tijd.

Na de aanname dat alle functies domein $\mathbb{N}^k$ en bereik $\mathbb{N}$ hebben, worden de functiesymbolen 0, S, $+$, $\cdot$, $\#$, $|x|$, $\lfloor \frac{1}{2}x \rfloor$ en het predikaatsymbool $\leq$ toegevoegd. Hierin is $\lfloor \frac{1}{2}x \rfloor$ het grootste gehele getal $\leq \frac{x}{2}$ en $x\#y = 2^{|x|\cdot|y|}$ de *smash*-functie.

Daarnaast zal er veelvuldig gebruik gemaakt worden van *kwantificatie*, om functies en predicaten te onderscheiden.
Kwantificatie houdt in dat er een n-plaatsig predicaat gevormd kan worden van een $n+1$-plaatsig predicaat. Hier is een predicaat een functie met bereik $\{0,1\}$.[1]

Definitie 6. [1],[4] Een *begrensde kwantor* is een kwantor van de vorm $(\forall x \leq t)$ of $(\exists x \leq t)$ met t een term die x niet bevat. Een *strikt begrensde kwantor* is een begrensde kwantor van de vorm $(\forall x \leq |t|)$ of $(\forall x \leq |t|)$, met t een term die x niet bevat. $(\forall x)$ en $(\exists x)$ zijn *onbegrensde kwantoren*.

De kwantificatie die van belang is, is de begrensde kwantificatie. Er wordt onderscheid gemaakt in *polynomiaal* begrensde kwantificatie, met grenzen van de vorm $2^{p(|t|)}$, en *logaritmisch* begrensde kwantificatie, met grenzen van de vorm $p(|t|)$.

2.1 De polynomiale hiërarchie

Nu volgt de definitie van de polynomiale hiërarchie, zoals Buss [4] hem in zijn proefschrift heeft gedefinieerd. Deze verschilt van andere definities omdat er naast functies ook predicaten in voorkomen.
Hierin zijn de verschillende niveau's gedefinieerd door de volgende termen.

Definitie 7. [3] Als Ψ een verzameling predicaten is, dan is $PB\exists(\Psi)$ ("*Polynomially Bounded Quantification*") de verzameling van predicaten A die uit te drukken zijn als

$$x \in A \Leftrightarrow (\exists y \leq 2^{p(|x|)})B(x,y),$$

met polynoom p en $B \in \Psi$.
Op dezelfde manier wordt $PB\forall(\Psi)$ gedefinieerd.

Definitie 8. [4] Als Ψ een verzameling predicaten is, dan is $LB\exists(\Psi)$ ("*Logarithmically Bounded Quantification*") de verzameling van predicaten A die uit te drukken zijn als

$$x \in A \Leftrightarrow (\exists y \leq p(|x|))B(x,y),$$

met polynoom p en $B \in \Psi$.
Op dezelfde manier wordt $LB\forall(\Psi)$ gedefinieerd.

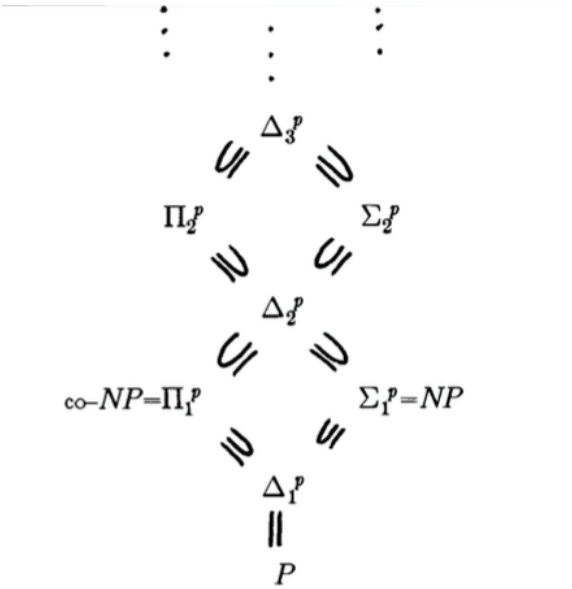
Het blijkt dat de polynomiaal begrensde kwantificatie en de logaritmisch begrensde kwantificatie correspondeert met respectievelijk de begrensde kwantificatie en de strikt begrensde kwantificatie. [4]

Definitie 9. [3] Als Ψ een verzameling predicaten is, zijn P^Ψ en FP^Ψ respectievelijk de verzamelingen predicaten en functies die in polynomiale tijd herkenbaar zijn met orakels[2] voor een eindig aantal predicaten in Ψ.

Met behulp van deze definities wordt de polynomiale hiërarchie als volgt gedefinieerd.

[1] Ook wel *False* of *True* voor respectievelijk 0 en 1.
[2] Een orakel betekent dat de machine een 'magisch'antwoord heeft voor die predicaten.

Figuur 1: De polynomiale hiërarchie.



Definitie 10. [3],[4] *De polynomiale hiërarchie.* In Figuur 1 is te zien hoe deze klassen in relatie staan tot elkaar.

- $[]_0^p$, de kleinste verzameling van functies die B[3] bevat, en gesloten onder compositie, $LB\exists$ en $LB\forall$;
- $\Delta_0^p = \Sigma_0^p = \Pi_0^p = PRED([]_0^p)$, de verzameling van predicaten herkenbaar in polynomiale tijd;
- $[]_{k+1}^p = FP^{\Sigma_k^p} = FP^{\Pi_k^p}$;
- $\Delta_{k+1}^p = PRED([]_{k+1}^p)$, de verzameling van predicaten in $[]_{k+1}^p$;
- $\Sigma_{k+1}^p = PB\exists(\Delta_{k+1}^p)$;
- $\Pi_{k+1}^p = PB\forall(\Delta_{k+1}^p)$;
- $PH = \bigcup_k \Sigma_k^p$.

De verzameling predicaten Δ_1^p, Σ_1^p en Π_1^p worden in de computerwetenschap respectievelijk P, NP en co-NP genoemd.
Er zijn meerdere open vragen omtrent de polynomiale hiërarchie. De hiërarchie stort in als blijkt dat $\Sigma_k^p = \Sigma_{k+1}^p$. Anders zeggen we dat de hiërarchie echt is. Echter is de vraag of de hiërarchie instort of niet nog onbeantwoord. Andere open vragen zijn bijvoorbeeld of geldt dat $P = NP$, of $NP =$ co-NP.
De polynomiale hiërarchie in Definitie 10 is een combinatie van twee verschillende definities van Buss, om hem zo compleet mogelijk te definiëren.

2.2 Formules in begrensde rekenkunde

Een rekenkundige formule is een formule in de eerste-orde logica die de logische symbolen $\wedge, \vee, \neg, \exists, \forall, \supset$ en $=$, en de niet-logische symbolen $0, S, +, \cdot, \#, |x|, \lfloor \frac{1}{2}x \rfloor$, en $\leq$ bevat.
Deze symbolen hebben de betekenis die beschreven is aan het begin van dit hoofdstuk.
De begrensde en strikt begrensde kwantoren zijn gedefinieerd in Definitie 6. Een formule heet een *begrensde formule* als deze uitsluitend begrensde kwantoren bevat. De verzameling van deze formules wordt Δ_0 genoemd. De volgende hiërarchie in de rekenkunde wordt beschreven door het tellen van het aantal alternerende onbegrensde kwantoren, de begrensde kwantoren buiten beschouwing gelaten.

[3]Hier is B de verzameling zoals gedefinieerd in Definitie 1.

Definitie 11. [1] Voor $n \geq 0$, worden de klassen Σ_n en Π_n als volgt gedefinieerd:

1. $\Sigma_0 = \Pi_0 = \Delta_0$, de verzameling van formules met uitsluitend begrensde kwantoren,
2. Σ_{n+1} is de verzameling formules van de vorm $(\exists x)A$, waarbij $A \in \Pi_n$,
3. Π_{n+1} is de verzameling formules van de vorm $(\forall x)A$, waarbij $A \in \Sigma_n$.

Deze klassen vormen samen de *rekenkundige hiërarchie*. Vergelijkbaar kan er ook een hiërarchie van de formules in de begrensde rekenkunde worden gedefinieerd. Deze hiërarchie telt het aantal alternerende begrensde kwantoren, de strikt begrensde kwantoren buiten beschouwing gelaten.

Definitie 12. De volgende verzamelingen van formules zijn gedefinieerd door Buss [4], en gedefinieerd door inductie.

1. $\Pi_0^b = \Sigma_0^b = \Delta_0^b$ is de verzameling van formules met uitsluitend strikt begrensde kwantoren.
2. Σ_{k+1}^b is inductief gedefinieerd door
   (a) $\Sigma_{k+1}^b \supset \Pi_k^b$;
   (b) Als $A \in \Sigma_{k+1}^b$, dan ook $(\exists x \leq t)A$ en $(\forall x \leq |t|)A$;
   (c) Als $A, B \in \Sigma_{k+1}^b$, dan ook $A \wedge B$ en $A \vee B$;
   (d) Als $A \in \Sigma_{k+1}^b$ en $B \in \Pi_{k+1}^b$, dan zijn $\neg B$ en $B \supset A$ in Σ_{k+1}^b.
3. Π_{k+1}^b is inductief gedefinieerd door
   (a) $\Pi_{k+1}^b \supset \Sigma_k^b$;
   (b) Als $A \in \Pi_{k+1}^b$, dan ook $(\forall x \leq t)A$ en $(\exists x \leq |t|)A$;
   (c) Als $A, B \in \Pi_{k+1}^b$, dan ook $A \wedge B$ en $A \vee B$;
   (d) Als $A \in \Pi_{k+1}^b$ en $B \in \Sigma_{k+1}^b$, dan zijn $\neg B$ en $B \supset A$ in Π_{k+1}^b.
4. Σ_{k+1}^b en Π_{k+1}^b zijn de kleinste verzamelingen die voldoen aan $(1) - (3)$.

Dit is een hiërarchie van begrensde kwantoren. Deze hiërarchie is grotendeels analoog aan de rekenkundige hiërarchie, met het enige verschil het tellen van de begrensde kwantoren in plaats van de onbegrensde kwantoren. Merk op dat in het superscript een b is toegevoegd, van *bounded arithmetic*, om het onderscheid te maken tussen de hiërarchie in begrensde en onbegrensde kwantoren. Het bewijs van de volgende stelling kan gevonden worden in [4].

Stelling 13. De klassen Σ_k^p (respectievelijk Π_k^p) bestaan uit predicaten die gedefinieerd worden door formules in Σ_k^b (respectievelijk Π_k^b).[4]

3 Eerste-orde rekenkunde

Om een goed beeld te krijgen bij de rekenkunde en verschillende theorieën van de rekenkunde, worden voorbeelden gegeven van axioma's en een aantal verschillende fragmenten. Deze theorieën komen uit [1], waar de relaties tussen de verschillende theorieën ook worden bewezen.
Als eerst worden axioma's geïntroduceerd die verschillende fragmenten van de eerste-orde rekenkunde definiëren. De verschillende axioma's kunnen worden ingedeeld in sterkere fragmenten en zwakkere fragmenten. Al deze systemen maken gebruik van dezelfde logische en niet-logische symbolen als beschreven in Hoofdstuk 2.
De belangrijkste eigenschappen van functies zijn de eigenschappen die een functie *bewijsbaar totaal* maken. Het zal blijken dat deze functies een grote rol spelen in fragmenten van de eerste-orde rekenkunde. De definities van bewijsbaar totale functies komen niet altijd overeen. In [5] worden er net wat andere eisen gesteld aan een functie dan in [1]. Beiden zullen genoemd worden in 3.3.

[4] De p in het superscript staat voor de polynomiale hiërarchie, en de b voor de rekenkundige hiërarchie (van *bounded arithmetic*).

3.1 Zwakke deelsystemen van rekenkunde

De meest bekende zwakke deelsystemen van de rekenkunde zijn de theorieën Q en T. Robinsons theorie Q bevat de beschreven niet-logische symbolen 0, S, $+$ en $\cdot$, en wordt geaxiomatiseerd door de volgende zes axioma's:

$$
\begin{aligned}
&(\forall x)(\neg Sx \neq 0)\\
&(\forall x)(\forall y)(Sx = Sy \supset x = y)\\
&(\forall x)(x \neq 0 \supset (\exists y)(Sy = x))\\
&(\forall x)(x + 0 = x)\\
&(\forall x)(\forall y)(x + Sy = S(x + y))\\
&(\forall x)(x \cdot 0 = 0)\\
&(\forall x)(x \cdot Sy = x \cdot y + x)
\end{aligned}
$$

Na toevoeging van het axioma dat het ongelijkheidssymbool $\leq$ bevat: $x \leq y \leftrightarrow (\exists z)(x + y = z)$, ontstaat het verlengde van de theorie Q dat $Q_{\leq}$ heet.
De nog zwakkere theorie R heeft dezelfde taal als Q, en wordt gedefinieerd door de volgende axioma's:

$$
\begin{aligned}
&S^m 0 \neq S^n 0, n \neq m\\
&S^m 0 + S^n 0 = S^{m+n} 0\\
&S^m 0 \cdot S^n 0 = S^{m \cdot n} 0\\
&(\forall x)(x \leq S^m 0 \vee S^m 0 \leq x)\\
&(\forall x)(x \leq S^m 0 \leftrightarrow x = 0 \vee x = S0 \vee x = S^2 0 \vee \cdots \vee x = S^m 0)
\end{aligned}
$$

Hierin is S^n een kortere notatie van n keer de functie S toepassen: $S(S(S(\cdots(S0))))$.
Er geldt dat $Q \models R$. [1]
Deze theorieën maken, in tegenstelling tot de theorieën die hierna genoemd gaan worden, geen gebruik van inductie-axioma's.

3.2 Sterke deelsystemen van rekenkunde

Nu volgt een beschrijving van de sterkere deelsystemen van de rekenkunde. Deze worden gedefinieerd door het gebruik van de zogeheten inductie-axioma's, minimalisatie-axioma's of collectie-axioma's. In deze sterkere deelsystemen is hier de rekenkundige hiërarchie zoals gedefinieerd in Definitie 11 van belang. Daarnaast worden de volgende axioma's gebruikt.

Definitie 14. [1]

1. De *inductie-axioma's*: als Φ een verzameling van formules is, dan bestaat het Φ-IND axioma uit de formules $A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)(A(x)$, voor alle formules $A \in \Phi$.

2. De *minimalisatie-axioma's* heten voor de verzameling Φ van formules Φ-MIN, en bestaan uit alle formules $(\exists x)A(x) \supset (\exists x)(A(x) \wedge \neg(\exists y)(y < x \wedge A(y)))$, voor alle formules $A \in \Phi$.

3. De *collectie-axioma's* voor de verzameling van formules Φ heten Φ-REPL en bestaan uit de formules $(\forall x \leq t)(\exists y)A(x, y) \supset (\exists z)(\forall x \leq t)(\exists y \leq z)A(x, y)$, voor alle formules $A \in \Phi$.

Deze axioma's vormen de basis van de sterkere fragmenten van de rekenkunde. Het belangrijkste fragment is de theorie $I\Delta_0$, die bestaat uit de axioma's van $Q_{\leq}$ en Δ_0-IND. Deze theorie is een subtheorie van alle andere fragmenten. Dit betekent dat als een functie of predicaat bewezen en dus gebruikt kan worden in $I\Delta_0$, dat gelijk voor alle andere theorieën geldt.
Deze overige fragmenten zijn zwakker dan $I\Delta_0$ en heten $I\Sigma_n$, $L\Sigma_n$ en $B\Sigma_n$ voor de $I\Delta_0$ axioma's samen met respectievelijk de Σ_n-IND, Σ_n-MIN en Σ_n-REPL axioma's.
Nu bestaat er de *Peano rekenkunde*, PA, die gedefinieerd wordt door de axioma's van theorie Q samen met de inductie-axioma's voor alle eerste-orde formules. Bovenstaande beschreven theorieën zijn verschillende fragmenten van de Peano rekenkunde, en zijn gelinkt door onderstaande logische relaties.[5]

[5] Het bewijs van deze relaties is te vinden in [1], Hoofdstuk 1.2.9.

$$
\begin{array}{ccccccc}
I\Sigma_{n+1} & & & & & & \\
\Downarrow & & & & & & \\
B\Sigma_{n+1} & \Leftrightarrow & B\Pi_n & & & & \\
\Downarrow & & & & & & \\
I\Sigma_n & \Leftrightarrow & I\Pi_n & \Leftrightarrow & L\Sigma_n & \Leftrightarrow & L\Pi_n
\end{array}
$$

De theorieën waar wij in geïnteresseerd zijn, zijn de sterkste van bovenstaand geformuleerde fragmenten. Nu heeft Buss in [1] als hoofdtheorie $I\Delta_0$ behandeld. Fragmenten van het verlengde van deze theorie, $I\Delta_0 + \Omega_1$, zijn de theorieën S_2^i en T_2^i.[1] Dit zijn de theorieën waar wij in geïnteresseerd zijn.

3.3 Definities in de rekenkunde

De volgende definities zijn van belang om de taal van de rekenkunde te kunnen vergroten, door functie- en predicaatsymbolen toe te voegen. Deze functie- en predicaatsymbolen moeten bepaalde eigenschappen bezitten om gebruikt te kunnen worden in inductieformules.
De eigenschap die van belang is, is dat een functie *representeerbaar* of *bewijsbaar totaal* moet zijn in de rekenkundige theorie. Er zijn verschillende opvattingen over de precieze definitie hiervan. In [5] wordt de definitie als volgt gegeven.

Definitie 15. Een functie $f : \mathbb{N}^k \to \mathbb{N}$ heet Σ_1-representeerbaar in een rekenkundige theorie T als er een Σ_1-formule $A(x, y)$ is zo dat geldt:

1. Voor alle $x \in \mathbb{N}^k$, $y \in \mathbb{N}$ geldt $T \vdash A(x, y) \Leftrightarrow f(x) = y$,
2. Voor alle $x \in \mathbb{N}^k$ geldt, $T \vdash (\exists!y)A(x, y)$.

Deze functies worden ook wel *bewijsbaar* in een theorie T genoemd. Een nog specifiekere klasse van functies worden de *bewijsbaar totale* of *bewijsbaar recursieve* functies genoemd.

Definitie 16. Een functie $f : \mathbb{N}^k \to \mathbb{N}$ heet *bewijsbaar totaal* in een rekenkundige theorie T als hij Σ_1-representeerbaar is in T en bovendien geldt

1. $T \vdash (\forall x)(\exists!y)A(x, y)$.

Alle berekenbare functies zijn Σ_1-representeerbaar in een theorie, maar ze zijn niet allemaal bewijsbaar totaal.
In [1] wordt er geen onderscheid gemaakt tussen bewijsbaar en bewijsbaar totaal. Dus worden Definities 15 en 16 samengevoegd tot de volgende definitie.

Definitie 17. Een functie $f : \mathbb{N}^k \to \mathbb{N}$ heet Σ_1-representeerbaar in een rekenkundige theorie T als er een Σ_1-formule $A(x, y)$ is zo dat geldt:

1. Voor alle $x \in \mathbb{N}^k, y \in \mathbb{N}$ geldt $T \vdash A(x, y) \Leftrightarrow f(x) = y$,
2. Voor alle $x \in \mathbb{N}^k$ geldt, $T \vdash (\exists y)A(x, y)$,
3. $T \vdash (\forall x)(\forall y)(\forall z)(A(x, y) \wedge A(x, z) \supset y = z)$.

Merk op dat de definities vrijwel exact hetzelfde zijn. Van Oosten [5] maakt het onderscheid tussen bewijsbaar en bewijsbaar totaal, met als tweede eigenschap dat er precies één oplossing is. Buss [1] maakt geen onderscheid tussen bewijsbaar en bewijsbaar totaal, en stelt niet de eis dat er exact één oplossing moet zijn.
Vanaf nu zal Definitie 17 aangehouden worden.
De naam bewijsbaar recursief komt van het feit dat voor elke Turing machine M er een Σ_1-formule $T_M(x, w, y)$ is die uitdrukt: 'w codeert een berekening van M op input x met uitkomst y'. Deze functie is Σ_1-representeerbaar door de formule $(\forall x)(\exists!y)(\exists w)(T_M(x, w, y))$. Andersom kan voor elke zin $(\forall x)(\exists!y)\phi(x, y)$ met ϕ een Σ_1-formule, de functie $y = f(x)$ berekend worden door een Turing machine die een waarde zoekt voor y en de waarheid van $(\exists y)\phi(x, y)$ onderzoekt.
Voor predicaten moet er ook een eigenschap gelden voordat deze symbolen toegevoegd mogen worden.

Definitie 18. Een predicaat $R(x)$ heet Δ_0-representeerbaar als geldt $R(x) \leftrightarrow \phi(x)$, met ϕ een Δ_0-formule.

Om te kunnen stellen dat de axioma's beschreven in dit hoofdstuk een goede basis zijn om eigenschappen van de gehele getallen te beschrijven, moet men eerst alle basisfuncties bewijzen. Dit proces wordt *bootstrapping* genoemd. Dit houdt in dat na bewijzen van deze basisfuncties, al deze functies vrij gebruikt mogen worden in de theorie. De theorie $I\Delta_0$ is vergelijkbaar met de theorieën S_2^1 en T_2^1, en deze zullen dan dan ook uitgebreid behandeld worden in volgende hoofdstukken.

In [1] wordt de theorie $I\Delta_0$ volledig behandeld en worden alle functies en predicaten, die bewezen kunnen worden in deze subtheorie, genoemd.

4 Begrensde rekenkunde

In deze sectie wordt de begrensde rekenkunde uitgebreid behandeld. Om een goed beeld te geven van de begrensde rekenkunde zijn stellingen en definities uit verschillende stukken van Buss [1], [3], [4] samengevoegd. Het belangrijkste zijn de theorieën S_2^i en T_2^i die het hoofdonderwerp zijn in [2], [3] en [4]. Deze theorieën corresponderen, zoals zal blijken, met functies die berekenbaar zijn in polynomiale tijd.

Begrensde rekenkunde is een verzamelnaam voor een familie van zwakkere fragmenten van de Peano rekenkunde. De begrensde rekenkunde maakt uitsluitend gebruik van begrensde kwantoren in de inductie-formules. Deze rekenkunde is interessant omdat hij sterk gelinkt is met de polynomiale hiërarchie. De taal van begrensde rekenkunde bestaat uit de eerder genoemde logische symbolen $\vee, \wedge, \neg, \supset, =, \exists, \forall$ en de niet-logische symbolen $S, 0, +, \cdot, |x|, \lfloor \frac{1}{2}x \rfloor, \#, \leq$. De *smash*-functie is belangrijk in de begrensde rekenkunde. Deze heeft precies het juiste groeipercentage om functies in de polynomiale hiërarchie te beschrijven. Omdat $1\#x = 2^{|x|}$ en $|\lfloor \frac{1}{2}(x\#y) \rfloor| = |x| \cdot |y|$, kunnen we de $\#, \lfloor \frac{1}{2}x \rfloor, \cdot$ gebruiken om de term $2^{p(|x|)}$ uit te drukken, met p een polynoom. Iedere term in de begrensde rekenkunde is begrensd door een vergelijkbare term voor een geschikt polynoom p. [4]
Met $0, S, +, \cdot$ kunnen precies alle natuurlijke getallen uitgedrukt worden.
Nu is in de begrensde rekenkunde dezelfde hiërarchie gedefinieerd als beschreven in Hoofdstuk 2.2. Daarbij hebben begrensde kwantoren de volgende eigenschap voor kwantor-uitwisseling: [1]
Laat A een formule zijn, dan

$$\begin{aligned}(\forall x \leq |s|)(\exists y \leq t)A(x,y) \Leftrightarrow &(\exists w \leq (2s+1)\#(4(2t+1)^2)) \\ &(\forall x \leq |s|)(A(x,\beta(x+1,w)) \wedge \beta(x+1,w) \leq t).\end{aligned}$$

Deze eigenschap laat strikt begrensde kwantoren in niet-strikt begrensde kwantoren opnemen, bij aanwezigheid van de β-funtie. Dit is *Gödels* $\beta-$functie, zoals gedefinieerd in 2.

In de begrensde rekenkunde heeft Buss [1] een verzameling van *BASIC* axioma's gedefinieerd.

$a \leq b \supset a \leq Sb$	$\lvert a\rvert = \lvert b\rvert \supset a\#c = b\#c$
$a \neq Sa$	$\lvert a\rvert = \lvert b\rvert + \lvert c\rvert \supset a\#d = (b\#d) \cdot (c\#d)$
$0 \leq a$	$a \leq a + b$
$a \leq b \wedge a \neq b \leftrightarrow Sa \leq b$	$a \leq b \wedge a \neq b \supset S(2 \cdot a) \leq 2 \cdot b \wedge S(2 \cdot a) \neq 2 \cdot b$
$a \neq 0 \supset 2 \cdot a \neq 0$	$a + b = b + a$
$a \leq b \vee b \leq a$	$a + 0 = a$
$a \leq b \wedge b \leq a \supset a = b$	$a + Sb = S(a + b)$
$a \leq b \wedge b \leq c \supset a \leq c$	$(a + b) + c = a + (b + c)$
$\lvert 0\rvert = 0$	$a + b \leq a + c \leftrightarrow b \leq c$
$\lvert S0\rvert = S0$	$a \cdot 0 = 0$
$a \leq 0 \supset \lvert 2 \cdot a\rvert = S(\lvert a\rvert) \wedge \lvert S(2 \cdot a) = S(\lvert a\rvert)$	$a \cdot (Sb) = (a \cdot b) + a$
$a \leq b \supset \lvert a\rvert \leq \lvert b\rvert$	$a \cdot b = b \cdot a$
$\lvert a\#b\rvert = S(\lvert a\rvert \cdot \lvert b\rvert)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
$0\#a = S0$	$S0 \leq a \supset (a \cdot b \leq a \cdot c \leftrightarrow b \leq c)$
$a \leq 0 \supset 1\#(2 \cdot a) = 2 \cdot (1\#a) \wedge 1\#(S(2 \cdot a)) = 2 \cdot (1\#a)$	$a \neq 0 \supset \lvert a\rvert = S(\lvert \lfloor \frac{1}{2}a \rfloor \rvert)$
$a\#b = b\#a$	$a = \lfloor \frac{1}{2}b \rfloor \leftrightarrow 2 \cdot a = b \vee S(2 \cdot a) = b$

Tabel 1. De *BASIC* axioma's gedefinieerd door Buss [1]. Deze axioma's hebben dezelfde rol voor theorieën in de begrensde rekenkunde als de axioma's van Q voor de fragmenten $I\Sigma_n$ van de Peano rekenkunde hebben.

Naast deze axioma's zijn er ook nog de zogeheten inductie-axioma's, zoals eerder genoemd in Definitie 14. De axioma's heten PIND en LIND, de P en L respectievelijk voor 'polynomiale' en 'lengte' inductie. [3]

- Ψ-*IND*: $A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)(Ax)$, voor $A \in \Psi$,
- Ψ-*PIND*: $A(0) \wedge (\forall x)(A(\lfloor \frac{1}{2}x \rfloor) \supset A(x)) \supset (\forall x)A(x)$, voor $A \in \Psi$,
- Ψ-*LIND*: $A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)A(|x|)$, voor $A \in \Psi$.

Merk op dat Ψ-*IND* sterker is dan Ψ-*PIND*. Stel bijvoorbeeld dat $A(0)$ waar is en dat de vraag is of $A(100)$ waar is. Door middel van Ψ-*IND*, wordt $A(1)$ afgeleid van $A(0)$, $A(2)$ van $A(1)$, enzovoorts. Dit kost 100 stappen. Bij gebruik van Ψ-*PIND* wordt eerst $A(1)$ afgeleid, en daarna $A(3)$, $A(6)$, $A(12)$, $A(25)$, enzovoorts. Dit kost 7 stappen. Omdat de conclusies van de axioma's hetzelfde zijn, is de hypothese van Ψ-*PIND* sterker dan die van Ψ-*IND*, en dus zijn de Ψ-*PIND* axioma's zwakker. Daarnaast bestaat de functie $x \mapsto 2^x$ niet in de begrensde rekenkunde. Dus de conclusie $(\forall x)A(|x|)$ van $\Psi-$*LIND* is zwakker dan $(\forall x)A(x)$. [4]

Nu zijn er twee belangrijke deelsystemen in de begrensde rekenkunde. Deze deelsystemen maken gebruik van de taal beschreven in dit hoofdstuk.

Definitie 19. [4] De theorieën S_2^i en T_2^i bestaan uit de volgende axioma's.

1. S_2^i bestaat uit
   (a) De *BASIC* axioma's;
   (b) De Σ_i^b-*PIND* axioma's.
2. T_2^i bestaat uit
   (a) De *BASIC* axioma's;
   (b) De Σ_i^b-*IND* axioma's.
3. S_2 is $\bigcup_i S_2^i$.
4. T_2 is $\bigcup_i T_2^i$.
5. $S_2^{(-1)} = T_2^{(-1)}$ is de theorie met enkel de *BASIC* axioma's.

Het meest interessant zijn de theorieën S_2^i, omdat deze de mooiste eigenschappen hebben. Wat zal blijken is dat in deze theorie de bewijsbaar totale functies precies de polynomiaal berekenbare functies zijn. Hierdoor is de sterkte van het fragment S_2^1 vergelijkbaar met die van polynomiale tijd berekenbaarheid.
De begrensde rekenkunde is sterk genoeg om vele functies te beschrijven. Als een theorie een functie op een bepaalde manier kan representeren, dan kan de taal van deze theorie vergroot worden door een nieuw functiesymbool toe te voegen voor de representeerbare functie. De functies die het meest interessant zijn, zijn die gebruikt kunnen worden in de inductie-axioma's.
Deze functies moeten de eigenschap bezitten, net als in Definitie 17, dat ze bewijsbaar totaal zijn. Nu volgt de definitie zoals die geldt voor theorieën in de begrensde rekenkunde, zoals bijvoorbeeld S_2^i en T_2^i.

Definitie 20. [4] Een functie $f : \mathbb{N}^k \to \mathbb{N}$ heet Σ_i^b-*representeerbaar* in een theorie T van de begrensde rekenkunde als er een Σ_i^b-formule $A(x, y)$ is zo dat geldt:

1. Voor alle $x \in \mathbb{N}^k$, $y \in \mathbb{N}$ geldt $T \vdash A(x, y) \Leftrightarrow f(x) = y$,
2. Voor alle $x \in \mathbb{N}^k$ geldt, $T \vdash (\exists y)A(x, y)$,
3. $T \vdash (\forall x)(\forall y)(\forall z)(A(x, y) \wedge A(x, z) \supset y = z)$.

Deze eigenschappen zijn erg belangrijk. Zoals beschreven mogen deze functies worden toegevoegd aan de taal van een fragment in de begrensde rekenkunde, en hun symbolen kunnen zonder beperkingen worden gebruikt in de inductie axioma's.
Voor predicaten kan een zelfde soort definitie worden opgesteld.

Definitie 21. [4] Neem een theorie R en een formule A. A heet Δ_i^b ten opzichte van de theorie R, dan en slechts dan als er formules $B \in \Sigma_i^b$ en $C \in \Pi_i^b$ zijn zo dat $R \vdash A \leftrightarrow B$ en $R \vdash A \leftrightarrow C$.

Deze twee definities zijn precies de voorwaarden waarvoor een nieuwe functie of nieuw predicaat geïntroduceerd kan worden in een fragment van de begrensde rekenkunde, en de nieuwe symbolen in de inductie-axioma's kunnen worden gebruikt.
Merk op dat dit dezelfde manier van definiëren is zoals eerder in de theorie $I\Delta_0$, in Hoofdstuk 3.

5 S_2^1 en T_2^1

In [1] wordt de theorie $I\Delta_0$ uitgebreid behandeld. Een andere theorie die erg overeenkomt hiermee is S_2^i. Voor een duidelijk overzicht van de belangrijkste definities is gekeken naar artikelen [2], [3]. De echte uitgebreide behandeling van de twee belangrijkste subtheorieën komt uit [4]. Hier zijn ook alle bewijzen terug te vinden.

De twee theorieën lijken in vele opzichten op elkaar. Het voornaamste verschil is dat de theorieën S_2^i de mooiste eigenschappen hebben. Deze hiërarchie van theorieën blijkt precies gelinkt te zijn met de polynomiale hiërarchie. De theorieën T_2^i staan in relatie tot S_2^i, die aan het eind van dit hoofdstuk naar voren komt.

S_2^1 is gedefinieerd als de theorie die beschreven wordt door de *BASIC*-axioma's en de Σ_1^b-*PIND* axioma's. De theorie T_2^1 is gedefinieerd als de theorie die beschreven wordt door de *BASIX*-axioma's en de Σ_1^b-*IND* axioma's.

5.1 De theorie S_2^1

De theorie S_2^1 bestaat uit de *BASIC*-axioma's en de Σ_1^b-*PIND* axioma's, zoals beschreven in Definitie 19. Om te kunnen laten zien dat S_2^1 een sterk systeem is dat veel functies en predicaten kan definiëren, moet er een proces plaatsvinden dat *bootstrapping* wordt genoemd. Dit houdt in dat alle functies en predicaten juist gedefinieerd moeten worden in dit systeem, en daarnaast moet ook bewezen worden dat deze functies bewijsbaar totaal zijn in het systeem. Het werk van Buss [4] is gebaseerd op het werk van meerdere wiskundigen[6], en wordt in dit hoofdstuk als leidraad gebruikt. Het is van belang dat de functies en predicaten in dit systeem respectievelijk Σ_1^b- en Δ_1^b-gedefinieerd zijn, omdat precies die formules dan vrij gebruikt mogen worden.
Er volgt nu een herhaling van Definitie 20 in het geval van de theorie S_2^1.

Definitie 22. [2] Een functie $f : \mathbb{N}^k \mapsto \mathbb{N}$ is *bewijsbaar recursief* door S_2^1 als er een formule $A(x, y) \in \Sigma_1^b$ is, zodat

1. Voor alle $x \in \mathbb{N}^k$, $y \in \mathbb{N}$ geldt $S_2^1 \vdash A(x, y) \Leftrightarrow f(x) = y$,
2. Voor alle $x \in \mathbb{N}^k$ geldt, $S_2^1 \vdash (\exists y)A(x, y)$,
3. $S_2^1 \vdash (\forall x)(\exists! y)A(x, y)$.

Deze functies heten bewijsbaar recursief, dit houdt in dat er een Turing machine is, die altijd stopt binnen polynomiale tijd.
Op dezelfde manier geldt de definitie voor de predicaten.

Definitie 23. [2] Een predicaat $Q \subseteq \mathbb{N}$ is Δ_1^b ten opzichte van S_2^1 als er een Σ_1^b-formule A en een Π_1^b-formule B is die Q definiëren, zo dat A en B bewijsbaar equivalent zijn in S_2^1.

Merk weer op dat deze definitie analoog is aan Definitie 21. Deze predicaten worden dus *recursief bewijsbaar* of *bewijsbaar totaal* in S_2^1 genoemd.

Het *bootstrapping* bestaat uit het bewijzen dat functies en predicaten respectievelijk Σ_1^b-representeerbaar in en Δ_1^b zijn ten opzichte van S_2^1. Dit omdat deze functies en predicaten dan geïntroduceerd mogen worden in de theorie, en hun functie- en predicaatsymbolen vrij gebruikt mogen worden.[7]

[6] Ed Nelson en Wilkie-Paris werken met de theorie S_2 in plaats van S_2^1, en Wilmers behandelt een zwakker fragment van deze theorie.
[7] Al deze bewijzen staan uitgebreid beschreven in Buss [4]. Vanwege ruimtegebrek zullen alleen de belangrijkste hier genoemd worden.

Tijdens het *bootstrappen* van S_2^1 komen vele functies en predicaten aan bod. Om te kunnen bewijzen dat deze functies en predicaten Σ_1^b-representeerbaar in en Δ_1^b zijn ten opzichte van S_2^1, bewijst Buss [4] dat de Σ_1^b-LIND axioma's stellingen zijn van S_2^1, zodat deze ook gebruikt kunnen worden in de bewijzen. Daarna begint het echte proces.

De volgende functies en predicaten zijn Σ_1^b-representeerbaar in en Δ_1^b ten opzichte van S_2^1.

1. Het predicaat $<$ en de functies $max(a,b)$ en $min(a,b)$:

$$\begin{aligned} &a < b \Leftrightarrow a \leq b \wedge a \neq b \\ &c = max(a,b) \Leftrightarrow (c \geq a \wedge c = b) \vee (c \geq b \wedge c = a) \\ &c = min(a,b) \Leftrightarrow (c \leq a \wedge c = b) \vee (c \leq b \wedge c = a) \end{aligned}$$

2. De *predecessor* functie is de inverse van de successor functie en wordt gedefinieerd door

$$b = P(a) \Leftrightarrow (a = 0 \wedge b = 0) \vee Sb = a$$

3. Het predicaat $Power2(a) \Leftrightarrow S(|P(a)|) = |a|$ en daarbij ook de volgende formules

$$\begin{aligned} &Power2(a) \supset a \neq 0 \\ &Power2(a) \supset Power2(a+a) \\ &Power2(a) \wedge |a| \leq |c| \supset a \leq c \\ &Power2(a) \wedge Power2(b) \wedge |a| = |b| \supset a = b \\ &Power2(a) \wedge a \neq 1 \supset Power2(\lfloor \frac{1}{2} a \rfloor) \end{aligned}$$

4. Een functie voor machtsverheffen kan gedefinieerd worden door

$$c = Exp(a,b) \Leftrightarrow Power2(c) \wedge |c| = 1 + min(|b|, a)$$

of met de notatie $Exp(a,b) = 2^{min(|b|,a)}$. Belangrijk hierbij is de opmerking dat deze functie een begrensde machtsverheffing is.

5. Een functie voor modulo 2 wordt gegeven door

$$b = Mod2(a) \Leftrightarrow b + 2 \cdot \lfloor \frac{1}{2} a \rfloor = a$$

$Mod2(a)$ is nul als a even is, en één als a oneven is.

6. Met het predicaat $Decomp$ kunnen functies voor 'less significant part' en 'more significant part' worden gedefinieerd

$$\begin{aligned} &Decomp(a,b,c,d) \Leftrightarrow |c| \leq b \wedge d \cdot 2^{min(|a|,b)} + c = a \\ &c = LSP(a,b) \Leftrightarrow (\exists d \leq a) Decomp(a,b,c,d) \\ &d = MSP(a,b) \Leftrightarrow (\exists c \leq a) Decomp(a,b,c,d) \end{aligned}$$

7. Een functie voor de waarde van het symbool op de 2^b-e plek van de binaire representatie van a wordt gegeven door

$$c = Bit(b,a) \Leftrightarrow c = Mod2(MSP(a,b)).$$

Daarbij kan de belangrijke eigenschap bewezen worden in S_2^1:

$$|a| \geq |b| \wedge (\forall y \leq |a|)(Bit(y,a) = Bit(y,b)) \supset a = b,$$

die zegt dat de binaire representatie van een getal uniek is.

8. Een begrensde versie van aftrekken wordt gedefinieerd als

$$c = LENMINUS(a,b) \Leftrightarrow (b \leq |a| \wedge c + b = |a|) \vee (b \geq |a| \wedge c = 0),$$

dus een minfunctie met als domein erg kleine getallen.

9. Ook aftrekken zonder restricties kan gedefinieerd worden.

$$c = a \dot{-} b \Leftrightarrow a + b = c \vee (c = 0 \wedge a < b)$$

10. Met het predicaat $QuoRem$ kunnen nog twee functies worden gedefinieerd.

$$QuoRem(a, b, c, d) \Leftrightarrow (b = 0 \wedge c = 0 \wedge d = 0) \vee (d < b \wedge a = c \cdot b + d)$$
$$c = \lfloor a/b \rfloor \Leftrightarrow (\exists d \leq b) QuoRem(a, b, c, d)$$
$$d = Rem(a, b) \Leftrightarrow (\exists c \leq a) QuoRem(a, b, c, d)$$

11.

$$b|a \Leftrightarrow Rem(a, b) = 0 \wedge b \neq 0$$

12. Voor de even en oneven getallen worden de volgende predicaten geïntroduceerd.

$$Even(a) \Leftrightarrow Mod2(a) = 0$$
$$Odd(a) \Leftrightarrow Mod2(a) = 1$$

13. Voor het definiëren van coderingen voor getallen wordt gebruik gemaakt van $Comma$ en $Digit$.

$$Comma(b, a) \Leftrightarrow Even(b) \wedge Bit(b, a) = 1 \wedge Bit(Sb, a) = 0$$
$$c = Digit(b, a) \Leftrightarrow [Even(b) \wedge Bit(Sb, a) = 1 \wedge Bit(b, a) = c] \vee [(Odd(b) \wedge Bit(Sb, a) = 0) \wedge c = 2]$$

Al deze functies en predicaten zijn dus zonder restricties bruikbaar in S_2^1, en dat door alleen een verzameling van basisaxioma's en één inductie axioma te definiëren.
Omdat met de taal van deze theorie de natuurlijke getallen kan aanduiden, kan met behulp van deze functies nog veel meer worden beschreven.

Nadat bovenstaande functies en predicaten geïntroduceerd zijn in S_2^1, moeten er rijtjes gedefinieerd worden met variabele lengte. De functies en predicaten die nog gedefinieerd moeten worden zijn

$Seq(w)$	*true* dan en slechts dan als w een geldige rij is
$Size(w)$	het maximum van de lengtes van de invoer van w
$Len(w)$	het aantal elementen in w
$\beta(i, w)$	de waarde van het i-de element in w
$*$	een functie die een nieuw element aan het einde van de reeks toevoegt
$**$	een functie een concatenatie maakt van twee reeksen

Om $Len(w)$ en β te kunnen definiëren moet er geteld worden, om een element in de reeks te vinden. Dit leidt tot de volgende definitie.

Definitie 24. [4] Laat $A(z, y, x)$ een formule zijn. De functie $f(y, x)$ is gedefinieerd door *lengte-begrensd tellen* van A dan en slechts dan als f voldoet aan $f(y, x) = (\#z \leq |y|)A(z, y, x)$, waarbij $(\#z \leq t)(\cdots)$ betekent 'het aantal getallen $z \leq t$ zo dat $\cdots$ '.

Op een zelfde manier zou *begrensd tellen* gedefinieerd kunnen worden, het enige verschil zou dan zijn dat de grens t geen lengte is. Het is echter een open vraag of functies die gedefinieerd worden door begrensd tellen altijd in de polynomiale hiërarchie zitten. Het blijkt dat we functies gedefinieerd door het begrensde tellen niet mogen gebruiken in S_2^1.[4] Maar voor functies gedefinieerd door het lengte-begrensde tellen geldt wel dat ze bewijsbaar totaal zijn in S_2^1.

Stelling 25. [4] Laat $A(z, y, x)$ een predicaat zijn dat Δ_1^b is ten opzichte van S_2^1. Als f een functie is die gedefinieerd wordt door lengte-begrensd tellen van A, dan is de functie f Σ_1^b-representeerbaar in S_2^1.

Stelling 26. [4] De volgende functies zijn Σ_1^b-representeerbaar in S_2^1 en kunnen geïntroduceerd worden als functiesymbolen:

1. $f_1(x) = min\{t(y) : y \leq |s|\}$;
2. $f_2(x) = max\{t(y) : y \leq |s|\}$;

3. $f_3(x) = (\mu y \leq |s|)A(y)$.

Hierin zijn s en t termen en is A een Δ_1^b-formule.

Nu de theorie S_2^1 helemaal uitgewerkt is, wordt duidelijk dat dit een complete beschrijving van alle natuurlijke getallen kan geven. Met behulp van deze functies en predicaten kunnen in deze subtheorie enorm veel functies berekend worden, en wat zal blijken is dat dit ook precies de functies zijn die uitvoerbaar (*feasible*) berekenbaar zijn, dus berekenbaar in polynomiale tijd.

5.2 De theorie T_2^1

Net als voor S_2^1, moet ook de theorie T_2^1 het *bootstrapping* ondergaan. Na het definiëren van een aantal simpele functies, volgt dat T_2^1 alle Σ_1^b-PIND axioma's bewijst. Daaruit volgt $T_2^1 \supseteq S_2^1$, en een deel van de functies die genoemd zijn in 5.1 kunnen dan ook worden geïntroduceerd in T_2^1. De volgende functies kunnen geïntroduceerd worden in T_2^1: 1, 2, 3, 4 en 6 uit Hoofdstuk 5.1.

Definitie 27. Als Q en R theorieën zijn en elke bewijsbare formule van R is een bewijsbare formule van Q, dan schrijven we $Q \vdash R$.

Stelling 28. [3] Neem $i \geq 1$. T_2^i bewijst de Σ_1^b-PIND axioma's, dus $T_2^i \vdash S_2^i$.

In het bewijs van deze stelling wordt uitsluitend gebruik gemaakt van de Σ_i^b-LIND axioma's, waarmee de volgende stelling ook bewezen is.

Stelling 29. [4] Laat R_i de theorie S_2^1 zijn plus de Σ_i^b-LIND axioma's. Dan is R_i equivalent met S_2^i.

5.3 Overige axioma's in de begrensde rekenkunde

5.3.1 Collectie-axioma's

Een belangrijke eigenschap van de natuurlijke getallen is het collectie-axioma, zoals al eerder genoemd in Hoofdstuk 3. Dit axioma laat onbegrensde kwantoren verplaatsen buiten het bereik van de begrensde kwantoren. Het algemene axioma is $(\forall x \leq a)(\exists y)A \leftrightarrow (\exists t)(\forall x \leq a)(\exists y \leq t)A$. Voor de theorieën die hier behandeld worden zijn de begrensde kwantoren belangrijk en worden de strikt begrensde kwantoren over het algemeen genegeerd. De volgende definitie laat zien dat begrensde kwantoren buiten het bereik van strikt begrensde kwantoren geplaatst kunnen worden.

Definitie 30. [4] De Σ_i^b-collectie axioma's zijn de formules van de vorm

$$(\forall x \leq |t|)(\exists y \leq s)A(x,y) \leftrightarrow (\exists w \leq SqBd(s,t))(\forall x \leq |t|)(A(x,\beta(Sx,w)) \wedge \beta(Sx,w) \leq s),$$

waarin s en t willekeurige termen zijn die respectievelijk geen y en x bevatten, en A een Σ_i^b-formule is. $SqBd$ is een term die de smash-functie bevat.

Stelling 31. [4] Laat $i \geq 1$. Dan zijn de Σ_i^b-collectie axioma's stellingen van S_2^i.

Uit deze stelling kan afgeleid worden dat $S_2^i \vdash R_i$. Daarnaast bewijst Buss [4] dat $R_{i+1} \vdash S_2^i$.

5.3.2 Minimalisatie-axioma's

Met behulp van de minimalisatie-axioma's kan de begrensde rekenkunde ook worden geaxiomatiseerd. Deze axioma's zijn eerder kort genoemd in Hoofdstuk 3.

Definitie 32. [4] Neem Ψ een verzameling van formules. De $\Psi-MIN$ axioma's worden gegeven door

$$(\exists x)A(x) \supset (\exists x)[A(x) \wedge (\forall y \leq x)(y \neq x \supset \neg A(y))],$$

met hierin $A \in \Psi$.
De $\Psi-LMIN$ axioma's worden gegeven door

$$(\exists x)A(x) \supset A(0) \vee (\exists x)[A(x) \wedge (\forall y \leq \lfloor \frac{1}{2}x \rfloor)(\neg A(y))],$$

met $A \in \Psi$.

Deze axioma's kunnen als volgt gebruikt worden in de theorie S_2^1.

Stelling 33. Laat $i \geq 1$. In de theorie S_2^1 geldt dat

1. $\Sigma_i^b - MIN$ equivalent is aan $\Pi_i^b - IND$, en
2. $\Sigma_i^b - LMIN$ equivalent is aan $\Pi_i^b - PIND$.

Dit leidt ons tot de volgende stelling over de $\Sigma_i^b - MIN$ axioma's.

Stelling 34. Laat $i \geq 1$. De $\Sigma_i^b - MIN$ axioma's zijn stellingen van S_2^{i+1}.

Deze stellingen worden bewezen door Buss in [4].

5.4 De relatie tussen S_2^i en T_2^i

Uit de stellingen en hun bewijzen uit paragraaf 5.3.2 komt de relatie tussen S_2^i en T_2^i die al eerder genoemd werd in Hoofdstuk 5.2.

Gevolg 35. Als $i \geq 0$, dan $S_2^{i+1} \vdash T_2^i$.

De volgende stelling kan bewezen worden met bovenstaand gevolg en stelling 28.

Stelling 36. Voor alle $i \geq 0$ geldt $T_2^{i+1} \Rightarrow S_2^{i+1}$ en $S_2^{i+1} \Rightarrow T_2^i$.

Uit deze stelling volgt dat $S_2 \equiv T_2$. Dus de theorieën $S_2^1, S_2^2, S_2^3, \cdots$ vormen een hiërarchie van subtheorieën van T_2, en hun vereniging is precies T_2. Het is echter een open vraag of de volgende inclusies echt zijn, of dat de hiërarchie instort als blijkt dat de subtheorieën toch gelijk zijn aan elkaar.
Open vraag: Zijn de volgende inclusies echt?

$$S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq \cdots$$

6 De polynomiale hiërarchie en S_2^1

Nu alle belangrijke eigenschappen duidelijk zijn, kan naar de conclusie toegewerkt worden. Door de bevindingen in de laatste hoofdstukken te combineren met stellingen en gevolgen uit [2],[3],[4] wordt het antwoord op de vraag welke functies gedefinieerd kunnen worden door de theorie S_2^1 en wat dit betekent voor de rekentijd. Daarnaast wordt ook duidelijk welke functies en predicaten representeerbaar zijn in welke theorieën.
In dit hoofdstuk wordt exclusief gekeken naar de fragmenten die axiomiseerbaar zijn door *PIND*-axioma's, dus de theoriën S_2^i. Omdat in deze scriptie uitsluitend eerste-orde rekenkunde behandeld wordt, wordt alleen naar de eerste-orde theorie gekeken die beschreven is in Hoofdstuk 5.1, het geval dat $i = 1$, dus S_2^1.
Een functie of predicaat wordt berekenbaar in *polynomiale tijd* genoemd als er een Turing machine M en een polynoom $p(n)$ bestaan zodat M de functie berekent of het predicaat herkent, en dat M stopt in tijd $\leq p(n)$ voor alle input van lengte n.
Zoals eerder beschreven, kunnen de Δ_1^b-representeerbare predicaten en Σ_1^b-representeerbare functies (ofwel de bewijsbaar totale functies en predicaten) vrij in de fragmenten van begrensde rekenkunde worden gebruikt. Wat belangrijk is, is dat dit precies alle functies en predicaten betreft die respectievelijk berekenbaar en herkenbaar zijn in polynomiale tijd.
Beschouw nogmaals de polynomiale hiërarchie van predicaten en functies. De klassen van predicaten waren Σ_k^p, Π_k^p en Δ_k^p, waarin Σ_1^p de klasse NP en Δ_1^p de klasse P werden genoemd. In plaats van dat predicaten alleen de waarden $\{0, 1\}$ aan kunnen nemen, mogen ze vanaf nu alle waarden in $\mathbb{N}$ aannemen.

Definitie 37. De klassen van functies worden $[]_k^p$ genoemd. Dit zijn de functies die berekenbaar zijn door een Turing machine in polynomiale tijd, waarbij de machine een orakel heeft voor een predicaat in Σ_{k-1}^p. De klasse $[]_1^p$ is dan de verzameling van functies die berekenbaar zijn in polynomiale tijd.

Buss heeft in zijn proefschrift de volgende stelling bewezen [4].

Stelling 38. [4] Neem $k \geq 1$. Laat f een m-plaatsige $[]_k^p$-functie zijn. Dan is er een Σ_k^b-formule A zo dat

1. Voor alle $x \in \mathbb{N}^m$ geldt $S_2^k \vdash A(x,y) \Leftrightarrow f(x) = y$,
2. Voor alle x geldt $S_2^k \vdash (\exists y)A(x,y)$,
3. $S_2^k \vdash (\forall x)(\forall y)(\forall z)(A(x,y) \wedge A(x,z) \supset y = z)$.

Merk de relatie op tussen deze stelling en Definitie 20.
Deze stelling zegt dat de theorie S_2^k alle polynomiaal berekenbare functies ten opzichte van de Σ_{k-1}^p predicaten kan Σ_k^b-representeren. Voor het geval dat $k = 1$ zegt deze stelling dus dat alle polynomiaal berekenbare functies Σ_1^b-representeerbaar zijn in S_2^1. Ofwel, alle polynomiaal berekenbare functies zijn bewijsbaar totaal in S_2^1.
Voor de predicaten is er een zelfde soort stelling.

Stelling 39. [4] Neem ≥ 1. Laat Q een m-plaatsig Δ_k^p predicaat zijn. Dan zijn er formules $A \in \Sigma_k^b$ en $B \in \Pi_k^b$ zo dat

1. $S_2^k \vdash (\forall x)(A(x) \leftrightarrow B(x))$,
2. Voor alle $n \in \mathbb{N}^m$, $A(n) \Leftrightarrow B(n) \Leftrightarrow Q(n)$.

Merk hier de relatie op tussen deze stelling en Definitie 21.
Deze stelling zegt dat alle polynomiaal berekenbare predicaten Δ_b^k zijn ten opzicht van de theorie S_2^k. Voor het geval $k = 1$ zegt deze stelling dus dat alle in polynomiale tijd herkenbare predicaten Δ_1^b zijn ten opzichte van S_2^1. Ofwel, alle in polynomiale tijd herkenbare predicaten zijn bewijsbaar totaal in S_2^1.
Dit kan samengevat worden in het volgende gevolg.

Gevolg 40. Elke in polynomiale tijd berekenbare functie en elk in polynomiale tijd herkenbare predicaat kan met een gedefinieerd functie- of predicaatsymbool geïntroduceerd worden in S_2^1, en gebruikt worden in inductieformules.

Tot nu toe is duidelijk dat elke functie in $[]_1^p$ bewijsbaar totaal is in S_2^1. Nu volgt het omgekeerde, dat de hoofdstelling in de begrensde rekenkunde wordt genoemd [4].

Stelling 41. (De Hoofdstelling). Laat $i \geq 1$. Neem aan dat $S_2^i \vdash (\forall x)(\exists y)A(x,y)$, waar $A(x,y)$ een Σ_i^b-formule is en x en y de enige vrije variabelen in A zijn. Dan is er een term $t(x)$, een Σ_i^b-formule B en een functie f in $[]_1^p$ zo dat

1. $S_2^i \vdash (\forall x)(\forall y)(B(x,y) \supset A(x,y))$,
2. Voor alle n, $\mathbb{N} \models B(n, f(n))$,
3. $S_2^i \vdash (\forall x)(\exists y \leq t)B(x,y)$,
4. $S_2^i \vdash (\forall x)(\forall y)(\forall z)(B(x,y) \wedge B(x,z) \supset y = z)$.

Met hieruit het volgende gevolg.

Gevolg 42. Neem aan dat f een functie is die Σ_i^b-representeerbaar is door S_2^i. Dan is f een $[]_i^p-$functie.

Merk op dat dit het precies het omgekeerde is van Stelling 38.
Deze stelling met zijn gevolg zegt dat als een functie bewijsbaar totaal is in een subtheorie S_2^i, deze op het niveau $[]_i^p$ van de polynomiale hiërarchie zit.
Dit is het algemene geval voor alle subtheoriën S_2^i. Deze stellingen en gevolgen tezamen geven het belangrijke algemene gevolg.

Gevolg 43. Een functie f is Σ_i^b-representeerbaar in S_2^i dan en slechts dan als $f \in []_i^p$.

Wat blijkt, is dat bewijsbaar totaal zijn en in op een bepaald niveau zitten in de polynomiale hiërarchie dus twee kanten op gaat.

Dus voor het speciale geval van S_2^1 geldt dan het volgende.

Gevolg 44. De Σ_1^b-definieerbare functies van S_2^1 zijn precies de polynomiaal berekenbare functies.

Dit is het belangrijkste gevolg. Alle functies die beschreven zijn in Hoofdstuk 5.1, waren Σ_1^b-representeerbaar. Deze functies werden ook wel de *bewijsbaar totale* of *bewijsbaar recursieve* functies genoemd. Nu blijkt dat precies deze functies tot de klasse $[]_1^p$ behoren, de klasse in de polynomiale hiërarchie die P werd genoemd. Dus deze functies zijn precies de functies die berekenbaar zijn in polynomiale tijd.
Voor de predicaten geldt een soortgelijk gevolg.

Gevolg 45. Laat $A(a)$ een formule zijn zodat S_2^1 bewijst dat A equivalent is aan een Σ_1^b- en een Π_1^b-formule. Dan is $A(a)$ een in polynomiale tijd herkenbaar predicaat.

In andere woorden zegt deze stelling dat S_2^1 bewijst dat $A \in NP \cap$ co-NP. Dus elk precidaat dat S_2^1-bewijsbaar is in $NP \cap$ co-NP, behoort ook tot de klasse P.

In de volgende stelling met gevolg staat duidelijk dat de hiërarchie van functies en predicaten correspondeert met de polynomiale hiërarchie.

Stelling 46. (Buss [4])

1. Een functie f is Σ_1^b-defineerbaar in S_2^1 dan en slechts dan als f berekenbaar is in polynomiale tijd.
2. Laat $i \geq 1$. Elke Σ_i^b-definieerbare functie van S_2^i staat op de $i-$de plek, $[]_i^p$, van de polynomiale hiërarchie.

Met andere woorden, een functie is bewijsbaar totaal in deze theorie S_2^1, precies dan als deze functie berekenbaar is in uitvoerbare tijd. Daarnaast correspondeert elk niveau van de functies met hetzelfde niveau in de polynomiale hiërarchie.
Hetzelfde geldt voor de predicaten.

Gevolg 47. (Buss [4])

1. Een predicaat R is Δ_1^b ten opzichte van S_2^1 dan en slechts dan als R polynomiaal berekenbaar is.
2. Laat $i \geq 1$. Elk Δ_i^b-definieerbare predicaat van S_2^i staat op de $i-$de plek, Δ_i^p, van de polynomiale hiërarchie.

Dus voor de predicaten geldt dat een predicaat bewijsbaar totaal is in S_2^1, dan en slechts dan als deze in polynomiale tijd herkenbaar is. Daarnaast correspondeert ook deze hiërarchie van predicaten met plekken in de polynomiale hiërarchie.
Merk nu op dat dus de hiërarchie van subtheorieën van S_2^i precies correspondeert met de polynomiale hiërarchie, en dat is wat dit zo'n bijzonder fragment van de begrensde rekenkunde maakt.
Voor de theorieën T_2^i blijkt ongeveer een zelfde soort stelling te gelden. [1]

Stelling 48. Laat $i > 1$.

1. Elke functie $f \in []_i^p$ is Σ_i^b-representeerbaar in T_2^{i-1},
2. Elk predicaat $R \in \Delta_i^p$ is Δ_i^p ten opzichte van T_2^{i-1}.

Deze stelling volgt uit de eerder genoemde Stelling 36.

7 Conclusie

Aan het begin van deze scriptie werd de vraag gesteld: Gegeven een theorie R, welke functies kan R representeren? Of welke functiesymbolen mogen geïntroduceerd worden in R?

Om het antwoord op deze vraag te vinden zijn de theorieën S_2^1 en T_2^1 behandeld. In het laatste hoofdstuk werd duidelijk dat voor deze twee theorieën geldt dat zij de functies kunnen representeren die bewijsbaar totaal worden genoemd. Voor deze functies, waarvan een deel genoemd is in Hoofdstuk 5, geldt dat ze met hun functiesymbool geïntroduceerd mogen worden in deze theorieën.

Ook geldt voor de functies die bewijsbaar totaal zijn in S_2^1 dat ze berekenbaar zijn in polynomiale tijd. Dus deze functies zijn binnen afzienbare tijd te berekenen door een computer van nu.

Referenties

[1] **Samuel R. Buss**, *First-Order Proof Theory of Arithmetic*, Chapter II of Handbook of Proof Theory, (Buss, ed.), Elsevier 1998.

[2] **Samuel R. Buss**, *The Polynomial Hierarchy and Fragments of Bounded Arithmetic*, Extended Abstract, Princeton University, 1985.

[3] **Samuel R. Buss**, *Bounded Arithmetic and Propositional Proof Complexity*, University of California, San Diego, 1985.

[4] **Samuel R. Buss**, *Bounded Arithmetic*, PhD dissertation, Princeton University, 1985.

[5] **Jaap van Oosten**, *Introduction to Peano Arithmetic*, Utrecht University, 1999.