Berekenbaarheid en Logica

14 april, 2009

André Deutz
LIACS
Universiteit Leiden

Formalisaties van het intuitieve begrip berekenbaarheid

Registermachines (Shepherdson en Sturgis)

Recursieve functies van Kleene

Turingmachines

Productiesystemen van Post

Markov algoritmen

Herbrand-Gödel berekenbaarheid

λ-calculus van Church

… …

“Modulo” Turingmachines

Formalisatie I: registermachines

Registermachines (Shepherdson en Sturgis)

Zie: J.C. Shepherdson, H.E. Sturgis, Computability of Recursive Functions, Journal Association of Computing Machinery 10 (1963)

Formalisatie I: registermachines

Drie typen instructies:

(a)

i: r_j +1 → k

// instructie nummer i luidt: tel bij r_j 1 op en ga over tot instructie k

(b)

i: r_j ·/ 1 → k, l

// i luidt: indien rj==0, ga dan over tot instructie l; indien r_j >1, trek dan 1 van r_j af en ga over tot instructie k.

(c)

i: stop

// een machine kan meer dan één stopinstructie hebben

Formalisatie II: partieel recursieve functies

Definitie: De functie $h(x,y_1,\ldots, y_n)$ heet gedefinieerd door ***recursie*** uit functies $f(y_1,\ldots, y_n)$ en

$g(x,y,y_1,\ldots, y_n)$ als

$$h(0,y_1,\ldots, y_n) = f(y_1,\ldots, y_n)$$
$$h(x+1,y_1,\ldots, y_n) = g(x, h(x,y_1,\ldots, y_n), y_1,\ldots, y_n)$$

Formalisatie II: partieel recursieve functies

Definitie: een functie f heet gedefinieerd door ***compositie*** uit h, $g_1, \ldots, g_n$ als f voldoet aan:

$$f(y_1, \ldots, y_m) = h(\, g_1(y_1, \ldots, y_m), \ldots, g_n(y_1, \ldots, y_m)\,)$$

Formalisatie II: partieel recursieve functies

Definitie: een functie f heet ***primitief recursief*** als f gedefinieerd kan worden door eindig vaak recursie en compositie toe te passen, uitgaande van de functies S, N, U_i^n :

a) $S(x) = x+1$ (opvolger functie)

b) $N(x) = 0$ (nul functie)

c) $U_i^n(x_1,\ldots, x_n) = x_i$, $1 \leq i \leq n$ (projectie functies)

Formalisatie II: partieel recursieve functies

Definitie: (beperkte) minimalisatie. Laat $f(x_1,\ldots,x_n,y)$ een functie zijn zodat voor elk n-tuple $(x_1,\ldots,x_n)$ er een y bestaat zodanig dat $f(x_1,\ldots,x_n,y) = 0$. Dan is de functie g afgeleid van f dmv beperkte **minimalisatie**, als geldt $g(x_1,\ldots,x_n) = \mu y(f(x_1,\ldots,x_n,y)=0)$, waarin $\mu y(f(x_1,\ldots,x_n,y)=0$ de kleinste y voorstelt waarvoor $f(x_1,\ldots,x_n,y)=0$.

Definitie: een functie f heet (***algemeen) recursief*** als f gedefinieerd kan worden door eindig vaak recursie, compositie en beperkte minimalisatie toe te passen, uitgaande van de functies S, N, U_i^n

Formalisatie II: partieel recursieve functies

Laat f: $\mathbf{N}^{k+1} \rightarrow \mathbf{N}$ (hoeft niet totaal te zijn)

g: $\mathbf{N}^{k} \rightarrow \mathbf{N}$ is afgeleid van f dmv (onbeperkte) **minimalisatie**, als g als volgt gedefinieerd is:

$g(\mathbf{x}) = y$ desda $f(x,y)=0$ en voor alle $z<y$, $f(\mathbf{x},z)$ is gedefinieerd & $f(\mathbf{x},z) \neq 0$

($\mathbf{x}=(x_1,x_2,\ldots,x_n)$)

Formalisatie II: partieel recursieve functies

Definitie: een functie f heet ***partieel recursief*** als f gedefinieerd kan worden door eindig vaak recursie, compositie en (onbeperkte) minimalisatie toe te passen, uitgaande van de functies S, N, U_i^n.

Formalisatie II: partieel recursieve functies

Primitief recursief

uit initiële functies (N, S, U_i^n) en recursie en compositie

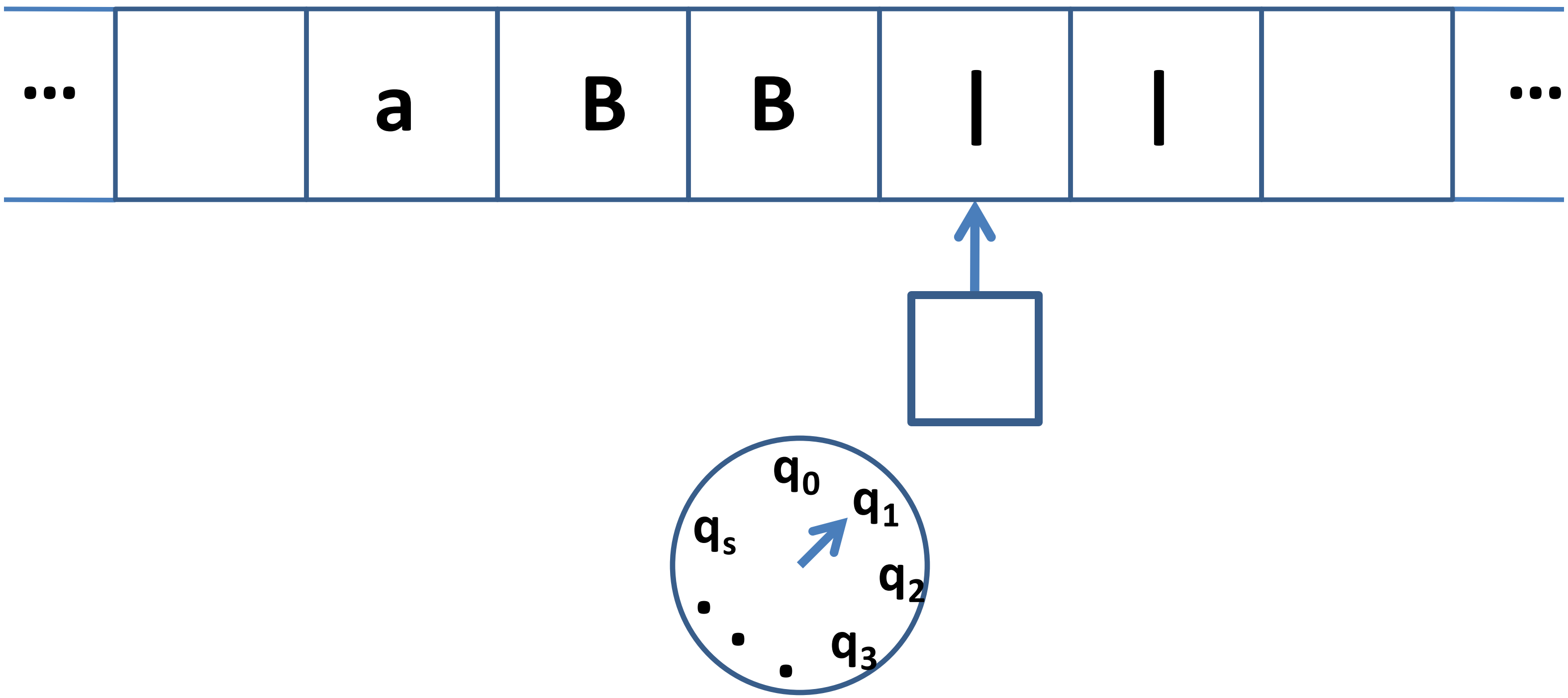
(Algemeen) recursief

uit initiële functies (N, S, U_i^n) en recursie en compositie en beperkte minimalisatie

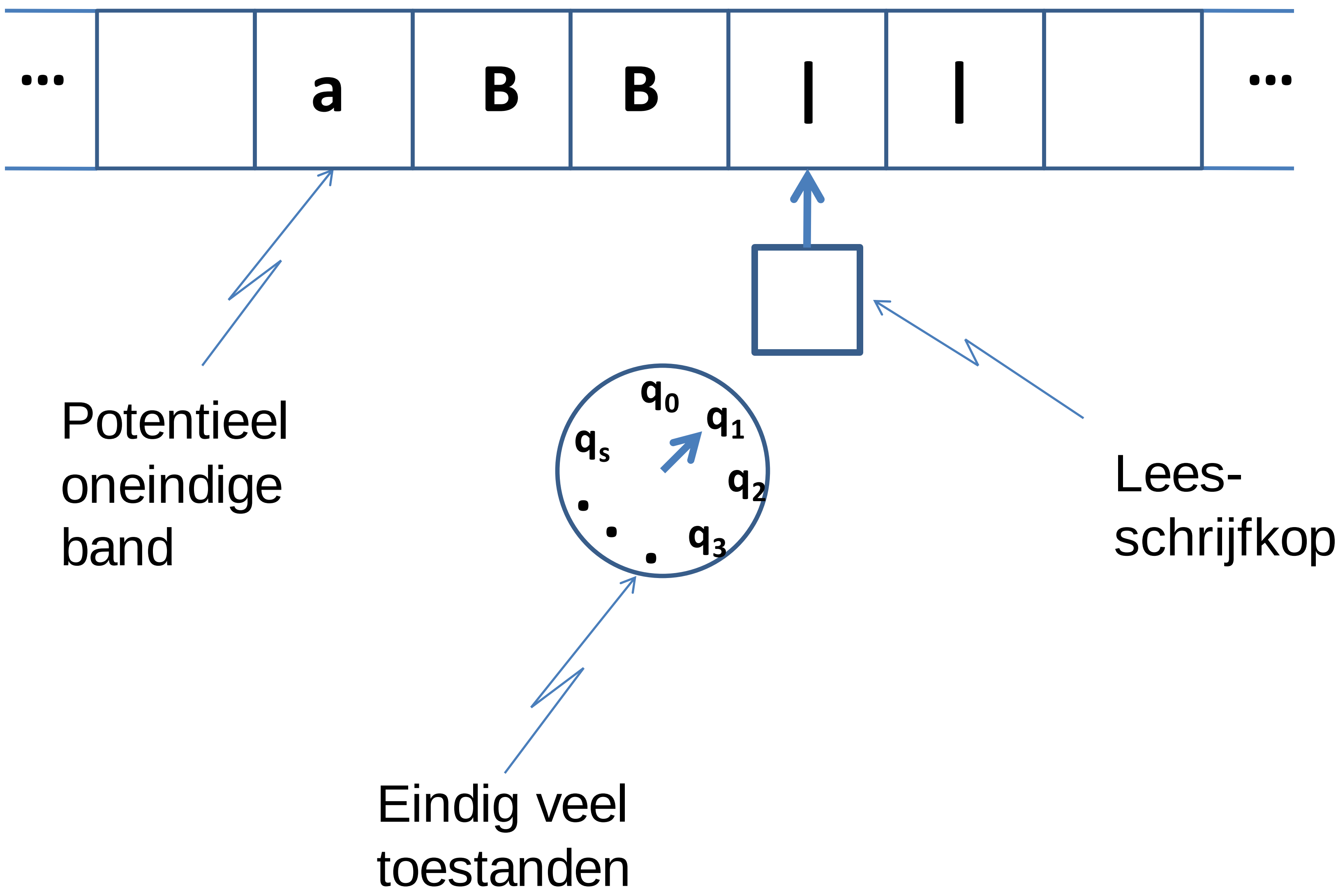
Partieel recursief

uit initiële functies (N, S, U_i^n) en recursie en compositie en onbeperkte minimalisatie

Formalisatie III: Turingmachines (1936)



Formalisatie III: Turingmachines (1936)



Formalisatie III: Turingmachines

- Bestaat uit een band die in vakjes verdeeld is en die aan beide zijden naar behoefte verlengd kan worden
- Kan symbolen uit een eindig alfabet in een vakje afdrukken of een vakje uitwissen.
- Heeft eindig veel inwendige toestanden.
- Leest (scant) op elk moment één vakje van de band
- Heeft de volgende elementaire handelingen:
  - In een bepaalde toestand een zeker symbool in het afgelezen vakje door een ander symbool vervangen en overgaan in een nieuwe toestand
  - In een bepaalde toestand een symbool aflezen en op grond daarvan naar links of rechts opschuiven, en overgaan in een nieuwe toestand
  - Het stopt

Formalisatie III: Turingmachines

- Formele definitie:

Een **Turingmachine** met een alphaber A bestaande uit bandsymbolen { $a_0,a_1,\ldots,a_n$ } en met interne toestanden { $q_0,q_1,\ldots,q_n$ } is een eindige verzameling $\mathcal{T}$ bestaande uit quadruples van de volgende vormen

- $q_j a_i a_k q_r$,
- $q_j a_i R q_r$,
- $q_j a_i L q_r$ zodat

- geen twee viertallen van $\mathcal{T}$ bevatten dezelfde twee eerste symbolen
- q_0 is de begintoestand en a_0 is gewoonlijk het blanco

Formalisatie III: Turingmachines

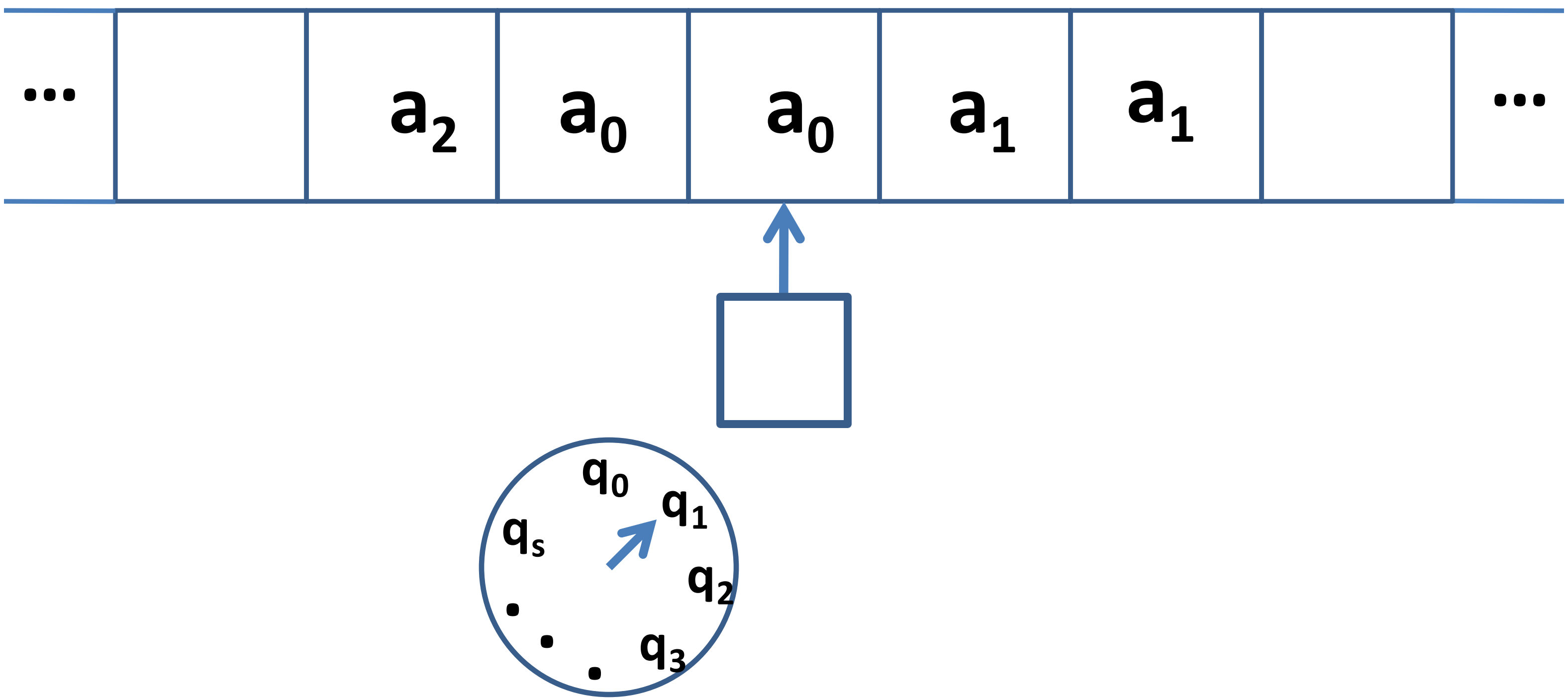
- **Bandbeschrijving (tape description, TD)** van $\mathcal{T}$ is een woord bestaande uit bandsymbolen behalve één; het enige niet-bandsymbool is een inwendige toestand q_i, en q_i is niet het laatste symbool van het woord.

- Het bandsymbool aan de naaste rechterkant van q_i is het bandsymbool dat op dit moment bekeken/gescand wordt.

- Als $q_i = q_0$, dan heet de bandbeschrijving de **initiële** bandbeschrijving

Formalisatie III: Turingmachines

Voorbeeld: $a_2a_0q_1a_0a_1a_1$ betekent dat de machine in de inwendige toestand q_1 is en het leeshoofd bekijkt het vakje aangeduid door de pijl



Formalisatie III: Turingmachines

De machine $\mathcal{T}$ **voert** een TD α **over in** TD β (notatie: $\alpha \rightarrow_{\mathcal{T}} \beta$) desda één van het volgende geldt:

1. α is van de vorm Pq_ja_iQ en β is van de vorm Pq_ra_kQ en $q_ja_ia_kq_r$ is een van de viertallen van $\mathcal{T}$
2. α is van de vorm $Pa_sq_ja_iQ$ en β is van de vorm $Pq_ra_sa_iQ$ en $q_ja_i\mathbf{L}q_r$ is een van de viertallen van $\mathcal{T}$
3. α is van de vorm q_ja_iQ en β is van de vorm $q_ra_0a_iQ$ en $q_ja_i\mathbf{L}q_r$ is een van de viertallen van $\mathcal{T}$ ("voeg blanco toe")
4. α is van de vorm $Pq_ja_ia_kQ$ en β is van de vorm $Pa_iq_ra_kQ$ en $q_ja_i\mathbf{R}q_r$ is een van de viertallen van $\mathcal{T}$
5. α is van de vorm Pq_ja_i en β is van de vorm $Pa_iq_ra_{k0}$ en $q_ja_i\mathbf{R}q_r$ is een van de viertallen van $\mathcal{T}$ ("voeg blanco toe")

Formalisatie III: Turingmachines

De machine $\mathcal{T}$ **stopt** bij een TD α desda er is ***geen*** TD β bestaat zodat $\alpha \rightarrow_{\mathcal{T}} \beta$

Opmerking: dit gebeurt natuurlijk wanneer $q_j a_i$ voorkomt in α maar $q_j a_i$ geen beginstuk(prefix) van een viertal van $\mathcal{T}$ vormt.

Formalisatie III: Turingmachines

Een **berekening** van $\mathcal{T}$ is een eindige reeks van TD's $\alpha_0,\ldots,\alpha_k$ ($k \geq 0$) zodat

1. α_0 is een initiële TD (i.e., de inwendige toestand die in α_0, voorkomt is q_0)
2. $\alpha_i \rightarrow_{\mathcal{T}} \beta_{i+1}$, voor $0 \leq i < k$
3. $\mathcal{T}$ stopt bij α_k

Deze berekening *begint* bij α_0 en *eindigt* bij α_k; we zeggen dat $\mathcal{T}$ op α_0 van toepassing is.

Formalisatie III: Turingmachines

Het **algoritme** $\mathcal{Alg}_{\mathcal{T}}$ dat door $\mathcal{T}$ bepaald wordt is als volgt gedefinieeerd:

Voor alle woorden P en Q over het alfabet A,

$\mathcal{Alg}_{\mathcal{T}}$(P) = Q desda er een berekening van $\mathcal{T}$ is dat begint met de TD q_0P en eindigt met een TD van de vorm $R_1q_jR_2$ waarin $Q=R_1R_2$.

Dit betekent dat als $\mathcal{T}$ begint aan de linker kant van P en er is niets anders op de band, dan stopt T met Q als de volledige inhoud van de band.

Voor sommige woorden P is $\mathcal{Alg}_{\mathcal{T}}$(P) niet gedefinieerd.

Formalisatie III: Turingmachines

Voorbeeld:

Laat $\mathcal{T}$ gegeven zijn door

$$q_0a_0Rq_0,\ \ q_0a_1Rq_1,\ \ q_0a_2Rq_1, \ldots ,q_0a_nRq_1$$

(a_0 is het blanco symbool)

Hoe ziet een berekening eruit?

Formalisatie III: Turingmachines

Arithmetische functies

Ipv a_1 schrijven we | en B ipv a_0.(B staat voor het blanco symbool.)

Voor een natuurlijk getal k, staat zijn **bandrepresentatie** $\tilde{k}$ voor het woord $|^{k+1}$ (i.e., het woord dat uit k+1 voorkomens van | bestaat).

$\tilde{0}$ = |; $\tilde{1}$ = ||; $\tilde{2}$=|||, etc.

De **bandrepresentatie** van een $(k_1,k_2,\ldots,k_n)\tilde{}$ van een n-tuple van natuurlijke getallen is het woord

$$B\,\tilde{k_1}\,B\,\tilde{k_2}\,B \ldots B\tilde{k_n}\,B$$

Voorbeeld: $(3,1,0,5)\tilde{}$ is B||||B||B|B||||||B.

Formalisatie III: Turingmachines

Een Turingmachine $\mathcal{T}$ berekent de volgende partiële functie $f_{\mathcal{T},1}$ van één variabele:

$f_{\mathcal{T},1}(k) = m$ desda het volgende geldt: $\mathcal{Alg}_{\mathcal{T}}(k\sim)$ is gedefinieerd en $\mathcal{Alg}_{\mathcal{T}}(k\sim) = E_1 m\sim E_2$, waarin E_1 en E_2 mogelijk leeg zijn of uitsluitend uit B's (blanco's) bestaan.

De functie $f_{\mathcal{T},1}$ is Turing berekenbaar. (Een 1-plaatsige functie partiële f is Turing berekenbaar desda er een Turingmachine $\mathcal{T}$ bestaat zodat $f = f_{\mathcal{T},1}$).

Formalisatie III: Turingmachines

Analoog voor meerplaatsige functies: voor elke n>1 berekent berekent een Turingmachine $\mathcal{T}$ een functie $f_{\mathcal{T},n}$ van n variabelen. Voor willekeurige natuurlijke getallen $k_1,k_2,\ldots,k_n$:

$f_{\mathcal{T},n}(k_1,k_2,\ldots,k_n)=m$ desda de volgende voorwaarde geldt: : $\mathcal{Alg}_{\mathcal{T}}((k_1,k_2,\ldots,k_n)^{\sim})$ is gedefinieerd en $\mathcal{Alg}_{\mathcal{T}}((k_1,k_2,\ldots,k_n)^{\sim}) = E_1 m^{\sim} E_2$, waarin E_1 en E_2 mogelijk leeg zijn of uitsluitend uit B's (blanco's) bestaan.

De partiële functie $f_{\mathcal{T},n}$ heet Turing berekenbaar. (Dus, een n-plaatsige functie f heet Turing berekenbaar desda er een Turingmachine $\mathcal{T}$ bestaat zodat $f = f_{\mathcal{T},n}$).

Formalisatie III: Turingmachines

Een Turingmachine $\mathcal{T}$ berekent de volgende partiële functie $f_{\mathcal{T},1}$ van één variabele:

$f_{\mathcal{T},1}(k) = m$ desda het volgende geldt: $\mathcal{Alg}_{\mathcal{T}}(k\sim)$ is gedefinieerd en $\mathcal{Alg}_{\mathcal{T}}(k\sim) = E_1 m \sim E_2$, waarin E_1 en E_2 mogelijk leeg zijn of uitsluitend uit B's (blanco's) bestaan.

De functie $f_{\mathcal{T},1}$ is Turing berekenbaar. (Een 1-plaatsige functie partiële f is Turing berekenbaar desda er een Turingmachine $\mathcal{T}$ bestaat zodat $f = f_{\mathcal{T},1}$).

Formalisatie III: Turingmachines

Opmerkingen

Aan het eind van de berekening van de waarde van Turing berekenbare functie, verschijnt alleen de waarde op de band, behalve wellicht blanco vakjes, en de locatie van de leeskop doet er niet toe.

In elk geval waarin de functie niet gedefinieerd is, zal de Turingmachine niet stoppen en als hij wel stopt dan heeft de band niet de geschikte vorm $E_1m{\sim}E_2$.

Formalisatie III: Turingmachines

Voorbeelden:

1) $\mathcal{T}$ met alfabet {B, | }, gedefinieerd door $q_0|Lq_1$, $q_1B|q_2$. $\mathcal{T}$ berekent de successor (opvolger) functie S(x). Dus de successor functie is Turing berekenbaar.

2) Wat berekent $\mathcal{T}$ met alfabet {B, | }, gedefinieerd door

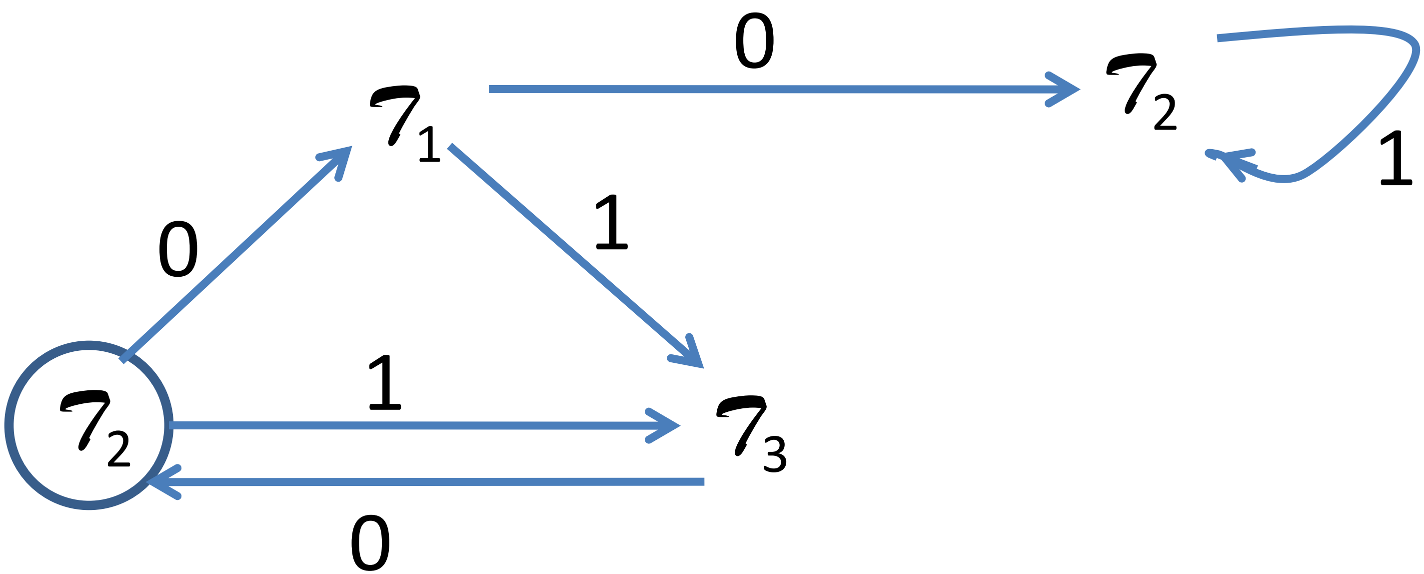
$$q_0|Bq_1,\ q_1BRq_0,\ q_0B|q_2\ ?$$

3) Optelling door Turingmachine met 7 transities:

$q_0|Bq_0$, q_0BRq_1, $q_1|Rq_1$, $q_1B|q_2$, $q_2R|q_2$, q_2BLq_3, $q_3|Bq_3$.

Formalisatie III: Turingmachines

Je kunt ook een higher-order formalisme (waarin je Turingmachines hergebruikt en combineert via bijvoorbeeld diagrammen) hanteren:



Formalisatie III: Turingmachines

Turingmachines kunnen Registermachines simuleren.

M registermachine met registers $R_1,\dots,R_n$ en instructies $1,\dots,k$

$\mathcal{T}_M$?

Met inhoud $r_1,\dots,r_n$ van de registers van M correspondeert de volgende TD:

$$|^{r_1+1}\ B\ |^{r_2+1}\ B\ |^{r_3+1}\ B\ \dots\ B\ |^{r_n+1}$$

$\mathcal{T}_M$ uit aantal $\mathcal{T}_i$; elke T_i imiteert het effect van instructie i van M

Instructie i op R_j $\Rightarrow$ $\mathcal{T}_i$ start in de eerste | van de j-de groep $|^{r_j+1}$

Formalisatie III: Turingmachines

Instructie i voert naar instructie j; instructie j werkt op R_l $\Rightarrow$ $\mathcal{T}_i$ stopt op de eerste | van de l-de groep $|^{r_l+1}$

$\mathcal{T}_i$ mbv hulpmachines $\mathcal{T}^R{}_t$, $\mathcal{T}^L{}_t$, $\mathcal{T}_l$, $\mathcal{T}_\ell'$

$\mathcal{T}^R{}_t$, werkt op een configuratie

$$|^{n_1+1}\ B \ \ldots\ B\ |^{n_t+1}\ B;$$

in startpositie leest $\mathcal{T}^R{}_t$ de eerste |; dan schuift $\mathcal{T}^R{}_t$, alle t groepen één positie naar rechts, en stopt op de laatste B

Formalisatie III: Turingmachines

$\mathcal{T}^{R}{}_{t}$:

$$\left.\begin{array}{l} q_{4k+1} \mid B\; q_{4k+2} \\ q_{4k+2}\; B\; R\; q_{4k+3} \\ q_{4k+3} \mid R\; q_{4k+3} \\ q_{4k+3}\; B \mid q_{4k+4} \\ q_{4k+4} \mid R\; q_{4k+5} \end{array}\right\} \; k=0,\ldots,t-1;\;\; q_{4t+1} \text{ is de stoptoestand}$$

Formalisatie III: Turingmachines

$\mathcal{T}^{L}_{t}$ werkt op een TD

$$|^{n_1+1}\ B\ \dots\ B\ |^{n_t+1}\ B;$$

in startpositie leest $\mathcal{T}^{L}_{t}$ de eerste | ;
schuift dan alle |'s één positie naar links en stopt dan op de laatste B

Formalisatie III: Turingmachines

$\mathcal{T}^{L}_{t}$:

$$\left.\begin{array}{l}
q_1 \mid L\ q_2 \\
q_{4k+2}\ B \mid q_{4k+3} \\
q_{4k+3} \mid R\ q_{4k+3} \\
q_{4k+3}\ B\ L\ q_{4k+4} \\
q_{4k+4} \mid B\ q_{4k+5} \\
q_{4k+5}\ B\ R\ q_{4k+6}
\end{array}\right\}\ k=0,\ldots,t-1;\ \ q_{4t+2} \text{ is de stoptoestand}$$

Formalisatie III: Turingmachines

$\mathcal{T}_l$:

$$\left.\begin{array}{l} q_{t+1}\ |\ R\ q_{t+1} \\ q_{t+1}\ B\ R\ q_{t+2} \end{array}\right\} \quad t=0,\ldots,l-1;$$

$q_{l+1}\ B\ L\ q_{l+2}$ $\quad$ q_{l+2} is de stoptoestand

$q_{l+1}\ |\ L\ q_{l+2}$

Begint vanuit de startpositie

$$|^{n_1+1}\ B\ \ldots\ B\ |^{n_l+1}\ B;$$

Ingesteld op de eerste |; schuift dan naar rechts over alle l blokken heen, eindigt op de laatste B.

Formalisatie III: Turingmachines

$\mathcal{T}_l$:

q1 B L q2

$$\left.\begin{array}{l} q_{t+2} \mid L\ q_{t+2} \\ q_{t+2}\ B\ L\ q_{t+3} \end{array}\right\} \quad t=0,\ldots,l-1;$$

q_{l+2} B R q_{l+3} $\qquad$ q_{l+4} is de stoptoestand

$q_{l+2} \mid R\ q_{l+3}$

q_{l+3} BR q_{l+4}

Begint vanuit de startpositie

$$|^{n_1+1}\ B\ \ldots\ B\ |^{n_l+1}\ B;$$

Ingesteld op de eerste |; schuift dan naar rechts over alle l blokken heen, eindigt op de laatste B.

Formalisatie III: Turingmachines

A) hulp machines $\mathcal{T}_j$:

i instructie van de vorm $\quad i\ r_k{+}1 \rightarrow j$

Instructie j werkt op R_l

$$\mathcal{T}': \begin{cases} q_i \mid R\ q_i \\ q_i\ B \mid q_i' \end{cases}$$

$\mathcal{T}_i$:

$$\mathcal{T}' \rightarrow \mathcal{T}^R{}_{n-k} \rightarrow \mathcal{T}'_{n-l+1}$$

q_i'

Eindtoestand van $\mathcal{T}'_{n-l+1}$ geïdentificeerd met q_j; alle andere toestanden verschillen van $q_1,\ldots,q_k$.

Formalisatie III: Turingmachines

B) hulp machines $\mathcal{T}_i$:

i instructie van de vorm i r_k •/ 1 → l, m (k < n)

Eerst $\mathcal{T}$'' met 2 eindtoestandedn; kijkt of in $|^{n_1+1}$, n1>0 ofwel n1 =0. in het tweede geval stopt hij in de toestand q'2 op de eerste en enige |, in het eerste geval verwijdert $\mathcal{T}$'' de laatste | uit het groepje en eindigt met $|^{n_1}$ BBq'$_7$.

De instructies l, m werken op R_s, R_t (respectievelijk).

Formalisatie III: Turingmachines

B) hulp machines $\mathcal{T}_i$:

i instructie van de vorm i r_k •/ 1 → l, m (k < n)

$$\left.\begin{array}{l}
q_i \mid R\ q'_1 \\
q'_1\ BL\ q'2 \\
q'_2 \mid R\ q'_3 \\
q'_3 \mid R\ q'_3 \\
q'_3\ B\ Lq'_4 \\
q'_4 \mid B\ q'_5 \\
q'_5\ B\ R\ q'_6 \\
q'_6\ BR\ q'_7
\end{array}\right\}\quad \mathcal{T}'''$$

Formalisatie III: Turingmachines

B) hulp machines $\mathcal{T}_i$:

i instructie van de vorm i r_k •/ 1 → l, m (k < n)

$\mathcal{T}_i$:

$$
\begin{tikzcd}
 & \mathcal{T}_{n-k+1} \arrow[r] & \mathcal{T}'_{n-t+1} \\
\mathcal{T}''' \arrow[ur, "q'_2"] \arrow[dr, "q'_7"'] & & \\
 & \mathcal{T}^{L}_{n-k} \arrow[r] & \mathcal{T}'_{n-s+1}
\end{tikzcd}
$$

eindtoestand van $\mathcal{T}'_{n-t+1}$ met q'_m indentificeren

eindtoestand van $\mathcal{T}'_{n-s+1}$ met q'_l indentificeren

Formalisatie III: Turingmachines

C) hulp machines $\mathcal{T}_i$:

i instructie van de vorm i r_n •/ 1 → l, m

$$\left.\begin{array}{l} q_i \mid R\, q'_1 \\ q'_1\, BB\, q'2 \\ q'_1 \mid R\, q'_3 \\ q'_3 BL\, q'_4 \\ q'_4 \mid B\, q'_4 \\ q'_4 \mid B\, q'_7 \end{array}\right\} \mathcal{T}'''$$

$\mathcal{T}_i$:

$$\begin{array}{l} \mathcal{T}''' \xrightarrow{q'_2} \mathcal{T}'_{n-t+1} \\ \mathcal{T}''' \xrightarrow{q'_7} \mathcal{T}'_{n-s+1} \end{array}$$

Formalisatie III: Turingmachines

D) De stop-instructies i_1, i_2 … corresponderen met de stoptoestanden q_{i_1}, q_{i_2}, … van de Turingmachine

$$\mathcal{T}_M = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \ldots \cup \mathcal{T}_k.$$

Formalisatie III: Turingmachines

Daar Registermachines alle partieel recursieve functies kunnen berekenen, kunnen de Turingmachines dit dus ook.

Formalisatie III: Turingmachines

Omgekeerd kunnen we ook laten zien dat elke Turing berekenbare functie een partieel recursieve functie is.

Hiertoe moeten de taal van Turing berekenbaarheid arithmetiseren door het toekennen van getallen, Goedel getallen, aan de expressies die in de studie van Turingmachines voorkomen.

Formalisatie III: Turingmachines

Symbool	Goedel getal
R	3
L	5
a_i	$7+4i$
q_i	$9+4i$

Bijvoorbeeld, het blanco symbool B (is a_0) krijgt 7; het stokje | (is a_1) krijgt 11.

Formalisatie III: Turingmachines

Een eindige reeks van symbolen $u_0,\ldots,u_k$ krijgt het Goedel getal:

$$p_0^{\,g(u_0)}\, p_1^{\,g(u_1)} \ldots\ldots p_k^{\,g(u_k)}$$

waarin $p_0,p_1,\ldots,p_k$ de priemgetallen 2,3,5,… in opklimmende orde zijn en $g(u_i)$ de Goedelgetallen van u_i zijn.

Bijvoorbeeld: $q_0a_0a_1q_0$ krijgt het Goedelgetal $2^9 3^7 5^{11} 7^9$.

Formalisatie III: Turingmachines

Een *expressie* dwz een eindige reeks van symbolen $u_0,\ldots,u_k$ krijgt het Goedel getal:

$$p_0^{\,g(u_0)}\, p_1^{\,g(u_1)} \ldots\ldots p_k^{\,g(u_k)}$$

waarin $p_0,p_1,\ldots,p_k$ de priemgetallen 2,3,5,… in opklimmende orde zijn en $g(u_i)$ de Goedelgetallen van u_i zijn.

Bijvoorbeeld: $q_0a_0a_1q_0$ krijgt het Goedelgetal $2^9 3^7 5^{11} 7^9$.

Formalisatie III: Turingmachines

Op een analoge manier kunnen we aan een eindige reeks van expressies $E_0,\ldots,E_k$ een Goedel getal toekennen:

$$p_0^{\,g(E_0)}\, p_1^{\,g(E_1)} \ldots\, p_k^{\,g(E_k)}$$

waarin $p_0,p_1,\ldots,p_k$ de priemgetallen 2,3,5,… in opklimmende orde zijn en $g(E_i)$ de Goedelgetallen van E_i zijn.

Formalisatie III: Turingmachines

Merk op:

Goedelgetal van een symbool is *oneven*

Goedelgetal van een expressie is *even*

Goedelgetal van een expressie is heeft voor p_0 een *oneven* exponent

Goedelgetal van een eindig rijtje expressies heeft voor p_0 een *even* exponent

Formalisatie III: Turingmachines

Definitie van de volgende functies:

$$\mathrm{len}(x), \quad (x)_j, \quad x*y$$

Laat x het Goedel getal zijn van een eindig rijtje $w_0,\ldots,w_k$; dwz $x = p_0^{\,g(w_0)}\, p_1^{\,g(w_1)} \ldots.\, p_k^{\,g(w_k)}$ waarin $g(w_i)$ het Goedelgetal is van w_i. Dan:

len(x) := k+1, de lengte van het rijtje

$(x)_j$ $:= g(w_j)$

Als y bovendien het Goedelnummer van het rijtje $v_0,v_1,\ldots,v_m$, dan is $x*y$ het Goedelnummer van de juxtapositie van de twee rijtjes: $w_0,w_1,\ldots,w_k,v_0,\ldots,v_m$.

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

IsToestandSym(x): x is het Goedelgetal van een toestandsymbool $(\exists u)_{u<x}(x=9+4u)$

IsAlfSym(x): x is het Goedelgetal van een alfabetsymbool $(\exists u)_{u<x}(x=7+4u)$

Quad(x): x is het Goedelgetal van een Turingmachine quadruple

$(\, len(x)=4 \wedge IsToestandSym((x)_0) \wedge IsAlfSym((x)_1) \wedge [IsAlfSym((x)_2) \vee (x)_2=3 \vee (x)_2=5] \wedge IsToestandSym((x)_3) \,)$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

TM(x): is het Gödelgetal van een Turingmachine (die gegeven is in de vorm van een eindig rijtje van legitieme quadruples):

$(\forall u)_{u<len(x)}Quad((x)_u)\wedge x>1\wedge(\forall u)_{u<len(x)}(\forall v)_{v<len(x)}(u\neq v \Rightarrow[((x)_u)_0 \neq ((x)_v)_0 \ \vee((x)_u)_1 \neq ((x)_v)_1])$

TD(x): x is het Gödelgetal van een bandbeschrijving

$x>1 \wedge(\forall u)_{u<len(x)}[IsAlfSym((x)_u)\vee IsToestandSym((x)_u))]\wedge (\exists_1 u)_{u<len(x)}IsToestandSym((x)_u)\wedge$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

TD(x): x is het Gödelgetal van een bandbeschrijving

$x>1 \wedge (\forall u)_{u<len(x)}[IsAlfSym((x)_u) \vee IsToestandSym((x)_u))] \wedge (\exists_1 u)_{u<len(x)} IsToestandSym((x)_u) \wedge$
$(\forall u)_{u<len(x)}(IsToestandSym((x)_u) \Rightarrow u+1<len(x))$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

Cons(x,y,z): x en y zijn Gödelgetallen van bandbeschrijvingen α en β en z is het Gödelgetal van Turingmachine die α in β transformeert.

$TD(x) \wedge TD(y) \wedge Quad(z) \wedge \ldots$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

NumTD(x): x is het Goedelgetal van een numerieke bandbeschrijving (i.e., een TD waarin de band de vorm

$E_1 k\tilde{\ } E_2$ waarin E_1 en E_2 of leeg zijn of uitsluitend uit blanco's bestaan en de plaats van de leeskop willekeurig is

$TD(x) \wedge (\forall u)_{u<len(x)}(IsAlfSym((x)_u) \Rightarrow (x)_u=7 \vee (x)_u=11) \wedge (\forall u)_{u<len(x)}(\forall v)_{v<len(x)} (\forall w)_{w<len(x)}(u<v \wedge v<w \wedge (x)_u=11 \wedge (x)_w=11 \Rightarrow (x)_u \neq 7)$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

Stop(x,z): z is het Gödelgetal van een Turingmachine $\mathcal{T}$ and x is het Goedelgetal van een TD waarop $\mathcal{T}$ stopt (“niet meer verder kan”)

$$TM(z) \wedge TD(x) \wedge \neg(\exists u)_{u<len(x)}[IsToestandSym((x)_u) \wedge (\exists u)_{v<len(z)}(\,((z)_v)_0 = (x)_u \wedge ((z)_v)_1 = (x)_{u+1}\,)]$$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

Comp(y,z): z is het Gödelgetal van een Turingmachine $\mathcal{T}$ and y is het Goedelgetal van een berekening van $\mathcal{T}$

$y>1 \wedge TM(z) \wedge (\exists u)_{u<len(y)} TD((y)_u) \wedge Stop((y)_{len(y) \dot{-} 1}), z) \wedge \ldots$

Num(x): x is het Goedelgetal van het woord x~ (i.e., van $|^{x+1}$): $Num(x) = \Pi_{u \leq x} (p_u)^{11}$

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

TR$(x_1,\ldots,x_n)$: het Gödelgetal van

de bandrepresentatie $(x_1,\ldots,x_n)$**~** van de n-tuple $(x_1,\ldots,x_n)$.

$TR(x_1,\ldots,x_n)=Num(x_1)*2^7*Num(x_2)*2^7*\ldots*2^7*Num(x_n)$

U(y): als y het Goedelgetal van een berekening is dat in numerieke bandbeschrijving resulteert, dan is U(y) het getal dat wordt gerepresenteerd op de laatste bandschrijving.

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

T_n $(z, x_1,\ldots,x_n,y)$: het Gödelgetal van een berekening van een Turingmachine met Gödelgetal z zodat de berekening begint op de band $(x_1,\ldots,x_n)$**~** met het lezen van de eerste | in x_1**~**, en eindigt met een numerieke TD.

Formalisatie III: Turingmachines

(Stelling) De volgende getal-theoretische relaties(predikaten) en functies zijn *primitief recursief*:

$T_n(z, x_1,\ldots,x_n,y)$: het Gödelgetal van een berekening van een Turingmachine met Gödelgetal z zodat de berekening begint op de band $(x_1,\ldots,x_n)^{\sim}$ met het lezen van de eerste | in $x_1^{\sim}$, en eindigt met een numerieke TD.

$\mathrm{Comp}(y,z) \wedge (y)_0 = 2^9 * \mathrm{TR}(x_1,\ldots,x_n) \wedge \mathrm{NumTD}((y)_{\mathrm{len}(y)\, \dot{-}\, 1})$

Formalisatie III: Turingmachines

(Stelling) Als een Turingmachine $\mathcal{T}$ een numerieke functie $f(x_1,x_2,\ldots, x_n)$ berekent en e is het Gödelgetal van $\mathcal{T}$, dan

$$f(x_1,x_2,\ldots, x_n) = U(\mu y T_n(e, x_1,\ldots,x_n,y))$$

Formalisatie III: Turingmachines

Bewijs:

laat $f(k_1,k_2,\ldots, k_n)$ natuurlijke getallen zijn. Dan is $f(k_1,k_2,\ldots, k_n)$ gedefinieerd desda er een berekening is van $\mathcal{T}$ die begint met $(k_1,k_2,\ldots, k_n)^{\sim}$ i.e., desda $(\exists y)T_n(e,k_1,k_2,\ldots, k_n,y)$. Dus, $f(k_1,k_2,\ldots, k_n)$ is gedefinieerd desda $\mu y T_n(e,k_1,k_2,\ldots, k_n,y)$ is gedefinieerd. Bovendien, als $f(k_1,k_2,\ldots, k_n)$ gedefinieerd is, dan is $\mu y T_n(e,k_1,k_2,\ldots, k_n,y)$ het Goedelgetal van berekening van $\mathcal{T}$ die begint met $(k_1,k_2,\ldots, k_n)^{\sim}$ en $U(\mu y T_n(e,k_1,k_2,\ldots, k_n,y))$ is het resultaat van de berekening, namelijk $f(k_1,k_2,\ldots, k_n)$

Formalisatie III: Turingmachines

(Stelling) Elke Turing berekenbare functie is partieel recursief.

Bewijs

Laat $f(x_1,x_2,..., x_n)$ een Turing berekenbaar zijn door een Turingmachine met Goedel nummer e, dan $f(x_1,x_2,..., x_n) = U(\mu y T_n(e, x_1,\ldots,x_n,y))$. Aangezien T_n primitief recursief is, is $\mu y T_n(e, x_1,\ldots,x_n,y)$ partieel recursief. Dus is $U(\mu y T_n(e, x_1,\ldots,x_n,y))$ partieel recursief.

Formalisatie III: Turingmachines

Stelling (Normal form stelling van Kleene):

Met het varieren van z over de natuurlijke getallen, somt $U(\mu y T_n(z, x_1,\ldots,x_n,y))$ alle partieel recursieve functies met n variabelen op (met herhalingen)

Onbeslisbare Problemen

Turing: onvolledigheid van Goedel als gevolg van niet-berekenbaarheid.

Som alle Turing berekenbare functies van 1 variabele op: $f_0(x)$, $f_1(x)$, $f_2(x)$, …

Definieer:

$$F(n) = \begin{cases} f_n(n) + 1, & \text{als } f_n(n) \text{ gedefinieerd is} \\ 0, & \text{als } f_n(n) \text{ niet gedefinieerd is} \end{cases}$$

Het is duidelijk dat F niet in de opsomming voorkomt, dus is F niet berekenbaar.

Onbeslisbare Problemen en Logica

Turing: onvolledigheid van Goedel als gevolg van niet-berekenbaarheid.

Onbeslisbare Problemen en Logica

Som alle Turing berekenbare functies van 1 variabele op: $f_0(x)$, $f_1(x)$, $f_2(x)$, …

Definieer:

$$F(n) = \begin{cases} f_n(n) + 1, & \text{als } f_n(n) \text{ gedefinieerd is} \\ 0, & \text{als } f_n(n) \text{ niet gedefinieerd is} \end{cases}$$

Het is duidelijk dat F niet als een f_k in de opsomming voorkomt, dus is F niet berekenbaar.

Dit betekent dat er geen effectieve procedure kan zijn die bepaalt of $f_n(n)$ gedefinieerd is of niet. (Onbeslisbaarheid halting problem.)

Onbeslisbare Problemen en Logica

Turing:

Dus er kan geen formele axiomatische theorie zijn die voor elk geval $f_n(n)$ laat bewijzen dat $f_n(n)$ gedefinieerd is of niet. Anders de bewijzen in een lijst zetten b_j en voor vaste n en f_n verifieren of vervolgens b_1 ,b_2 … een bewijs is voor "$f_n(n)$ is gedefinieerd" of "$f_n(n)$ is niet gedefinieerd". Maar dan is F berekenbaar. Tegenspraak.

Dus er zijn beweringen ϕ die waar $(I \models \phi)$ en

$S \nvdash \phi$ en $S \nvdash \neg\phi$

Onbeslisbare Problemen en Logica

Andere onbeslisbare problemen:

Geldigheid van formules in de predikatenlogica

Bewijsbaarheid van formules in de Peano rekenkunde

Waarheid (onder standaard interpretatie) van formules in de Peano-rekenkunde

Bestaan van nulpunten voor polynomen met gehele coefficienten