Berechenbarkeit

Predictability is like knowing the path a river takes. The river starts at its source and flows down to the sea. Along the way, it may turn, twist, and divide, but it always follows the path of least resistance due to gravity. Knowing the terrain allows us to predict where the river will go.
- ChatGPT

1 Berechenbarkeit

Konvention: Sprechen wir von einer $e \in \mathbb{N}_0$ oder $(e_1, \cdots, e_n) \in \mathbb{N}_0^n$ wobei $n \in \mathbb{N}$ als Eingabe für eine TM oder Ausgabe einer TM, so bedetet dies, dass die Eingabe bzw. Ausgabe $bin(e)$ bzw $(bin(e_1), \cdots, bin(e_n))$ ist. Dies erlaubt es über partiell berechnenbare Funktionene $\Phi : \mathbb{N}_0^n \rightsquigarrow \mathbb{N}_0$ wobei $n \in \mathbb{N}$ zu sprechen und $L \subseteq \mathbb{N}_0$ als Sprache über $\{0,1\}$ aufzufassen.

1.1 Definition (Code)

Wir betrachten die Funktion code (mit geeignetem Definitionsbereich) und Zielmenge $\{0,1\}^*$, für die folgendes gilt. Zunächst gelte

$$code(L) = 10 \qquad code(S) = 00 \qquad code(R) = 01$$

Für eine Instruktion $I = (q, a, q', a', B) \in \mathbb{N}_0 \times \{0,1\} \to \mathbb{N}_0 \times \{0,1\} \times \{L, S, R\}$ einer normierten TM sei

$$code(I) = 0^{|bin(q)|} 1 bin(q) a 0^{|bin(q')|} 1 bin(q') a' code(B)$$

Für eine endliche Menge $\Delta \subseteq \mathbb{N}_0 \times \{0,1\} \to \mathbb{N}_0 \times \{0,1\} \times \{L, S, R\}$ von Instruktionen einer normierten TM und $i \in [|\Delta|]$ sein $code_i(\Delta)$ dann ein längenlexikographische Ordnung i-te Wort in $\{code(I) : I \in \Delta\}$ und sei

$$code(\Delta) = code_1(\Delta), \cdots, code_{|\Delta|}(\Delta)$$

Für eine normierte TM $M = (\{0, \cdots, n\}, \{0,1\}, \{\square, 0, 1\}, \Delta, 0, \{0\})$ sei

$$code(M) = 0^{|bin(n)|} 1 bin(n) code(\Delta)$$

der **Code** von M. Relevant ist hierbei dass es eine geeignete effektive Codierung von Turingmachinen durch Binärwörter gibt, so dass folgendes gilt

- Jede normierte TM hat einen Code
- Keine zwei verschiedene normierten TMs haben den gleichen Code.
- Die Sprache der Codes von Turingmachinen ist entscheidbar
- Codes können eine geeignete Repräsentation der durch sie codierten TMs umgewandelt werden, die es insbesondere erlauben die codierten TMs effekiv zu simulieren.
- geignete Repräsentationen von TMs können effektiv in ihre Codes umgewandet werden.

1.2 Definition (standardaufzählung)

Sei $\hat{w_0}, \hat{w_1}, \cdots$ die Aufzählung aller Codes normierter TMs in längenlexikographischer Ordnung. Für $e \in \mathbb{N}_0$ sei M_e die durch $\hat{w_e}$ codierte TM und für $n \in \mathbb{N}$ sei $\Phi_e^n : \mathbb{N}_0^n \to \mathbb{N}_0$ die von M_e berechnete n-äre partielle Funktion. Für $n \in \mathbb{N}$ heißt die Folge (Φ_e^n) mit $e \in \mathbb{N}$ **standardaufzählung** der n-ären partiell berechenbaren Funktion. Für $n \in \mathbb{N}$ und eine partiell berechenbare n-äre Funktion $\varphi : \mathbb{N}_0^n \to \mathbb{N}_0$ heißt jede zahl $e \in \mathbb{N}_0$ mit $\Phi_e^n = \varphi$ **Index** von φ.

Konvention:

Ergibt sich n aus dem Kontext, so schreiben wir auch Φ_e statt Φ_e^n

1.3 Bemerkung

Für $n \in \mathbb{N}$ und eine partielle berechnbare n-äre partielle Funktion $\Phi : \mathbb{N}_0^n \to \mathbb{N}_0$ gibt es unendlich viele Indizes von φ.

1.4 Definition (U)

Es bezeichnet U die normierte TM, die bei Eingabe $(e,x_1,\cdots,x_n) \in \mathbb{N}_0^{n+1}$ wobei $n \in \mathbb{N}$ die normierte TM $\mathscr{M}_e$ bei Eingabe $(x_1,\cdots,x_n)$ simuliert und falls diese terminiert die Ausgabe der Simulierten ausgibt.

1.5 Definition (Universell)

Eine DTM U heißt **Universell**, wenn es für alle $n \in \mathbb{N}$ und alle partiell berechenbaren Funktionen $\varphi : \mathbb{N}_0^n \rightsquigarrow \mathbb{N}_0$ eine $e \in \mathbb{N}$, so dass

$$U(e,x_1,\cdots,x_n) = \varphi(x_1,\cdots,x_n)$$

$\forall x_1,\cdots,x_n \in \mathbb{N}_0$ gilt.

1.6 Bemerkung

Die TM U ist universell, denn für $e \in \mathbb{N}_0$, $n \in \mathbb{N}$ und $x_1,\cdots,x_n \in \mathbb{N}_0$ gilt

$$U(e,x_1,\cdots,x_n) = \Phi_e(x_1,\cdots,x_n)$$

$$(x,y) \mapsto x^y$$

$$y \mapsto 2^y$$

$$(x_1,\cdots,x_m,y_1,\cdots,y_n) \mapsto \varphi(x_1,\cdots,x_m,y_1,\cdots,y_n) partiell berechenbar$$

$$\rightsquigarrow (y_1,\cdots,y_m) \mapsto \varphi(x_1,\cdots,x_m,y_1,\cdots,y_n) partiell berechenbar$$

1.7 Satz (s_n^m - Theorem)

$\forall m,n \in \mathbb{N}$ existiert eine berechenbare Funktion $s_n^m : \mathbb{N}_0^{m+1} \to \mathbb{N}_0$ mit

$$\Phi_e^{m+1}(x_1\cdots,x_m,y_1,\cdots,y_n) = \Phi^n_{s_n^m(e,x_1,\cdots,x_m)}(y_1,\cdots,y_n)$$

$\forall e,x_1,\cdots,x_m,y_1,\cdots,y_n \in \mathbb{N}_0$

Beweis. Fixiere $m \in \mathbb{N}$. Betrachte die DTM S , die bei Eingabe $(e,x_1,\cdots,x_m) \in \mathbb{N}_0^{m+1}$ wie folgt vorfährt.

- Zunächst bestimmt S den Code von $\mathscr{M}_e$
- der Code von $\mathscr{M}_\urcorner$ wird dann in einen Code einer normierten TM $\mathscr{M}$ umgewandet, die zunächst $x_1\square\cdots\square x_m\square$ neben die Eingabe schreibt, dan den Kopf auf das erste Feld des beschriebenen Bandteilsbewegt und dann wie $\mathscr{M}_\urcorner$ arbeitet.
- Es wird bestimmt an welcher Stelle der Standardaufzählung der Code von auftaucht und diese Stelle wird ausgegeben.

Sei s_n^m die von S berechnete $(m+1)$-äre partielle Funktion. Dann ist s_n^m eine Funktion wie gewünscht. Es gibt überabzählbar viele Binärsprachen, denn: Betrachte Aufzählung von Binärsprachen $L_1,L_2,\cdots$

$$L \text{ mit } \mathbb{1}_L(i) = \begin{cases} 0, & \text{wenn } \mathbb{1}_{L_i}(i) = 1 \\ 1, & \text{wenn } \mathbb{1}_{L_i}(i) = 0 \end{cases}$$

□

$$\begin{array}{cccc} \mathbb{1}_{L_0}(0) & \mathbb{1}_{L_0}(1) & \mathbb{1}_{L_0}(2) & \mathbb{1}_{L_0}(3) \\ \mathbb{1}_{L_1}(0) & \mathbb{1}_{L_1}(1) & \mathbb{1}_{L_1}(2) & \mathbb{1}_{L_1}(3) \\ \mathbb{1}_{L_2}(0) & \mathbb{1}_{L_2}(1) & \mathbb{1}_{L_2}(2) & \mathbb{1}_{L_2}(3) \end{array}$$

Standardaufzählung

1.8 Definition (diagonales Halteproblem)

Die Menge $H_{diag} := \{e \in \mathbb{N}_0 : \Phi_e(e)\downarrow\}$ heißt **diagonales Halteproblem**.

Proposition 1.8.1. *Das diagonale Halteproblem ist rekursiv aufzählbar.*

Beweis. Die DTM, die bei Eingabe $e \in \mathbb{N}_0$ wie U bei Eingabe (e,e) arbeitet, aber bei terminieren 1 statt der Ausgabe von U ausgibt berechnet die partielle charachteristische Funktion von H_{diag}. Die partielle Funktion $x_{H_{diag}}$ ist also partiell berechenbar. Die partielle Funktion $x_{H^c_{diag}}$ ist nicht partiell berechenbar, dann: Betrachte Standardaufzählung □

$$\begin{array}{cccc} \Phi_{L_0}(0) & \Phi_{L_0}(1) & \Phi_{L_0}(2) & \Phi_{L_0}(3) \\ \Phi_{L_1}(0) & \Phi_{L_1}(1) & \Phi_{L_1}(2) & \Phi_{L_1}(3) \\ \Phi_{L_2}(0) & \Phi_{L_2}(1) & \Phi_{L_2}(2) & \Phi_{L_2}(3) \end{array}$$

Standardaufzählung

φ mit $\varphi(i) = \begin{cases} \uparrow, & \text{wenn } \Phi_i(i)\downarrow \\ \downarrow, & \text{wenn } \Phi_i(i)\uparrow \end{cases}$ Wird nicht aufgezählt.

1.9 Satz

Das diagonale Halteproblem ist nicht entscheidbar.

Beweis. Angenommen H_{diag} wäre entscheidbar. Dann wäre die partielle charakteristische Funktion φ von $H^c_{diag} = \mathbb{N}_0 / H_{diag}$ partiell berechenbar, es gäbe also ein Index $e \in \mathbb{N}_0$ von φ. Es folge

$$e \in H^c_{diag} \Leftrightarrow \varphi(e)\downarrow \Leftrightarrow \Phi_e(e)\downarrow \Leftrightarrow e \in H_{diag} \Leftrightarrow e \notin H^c_{diag}$$

Die ist ein Wiederspruch. □

1.10 m-Reduktion

Für eine Sprache A über einem Alphabet Σ und eine Sprache B über einem Alphabet Γ ist A genau dann **many-one-reduzierbar**, auch **m-reduzierbar**, auf B, kurz $A \leq_m B$, wenn es eine berechebare Funktion. $f : \Sigma^* \to \Gamma^*$ gibt so dass

$$w \in A \Leftrightarrow f(w) \in B$$

$\forall w \in \Sigma^*$ gilt. Gelten $A \leq_m B$ und $B \leq_m A$, so sind A und B **many-one-äquivalent** auch **m-äquivalent**, kurz $A =_m B$.

1.11 Bemerkung

(i) $\leq_m$ ist transitiv.

(ii) Gilt $A \leq_m B$ für Sprachen A und B und ist B entscheidbar, so ist auch A entscheidbar.

(iii) Alle entscheidbaren Sprachen L mit $\varnothing \neq L \neq \mathbb{N}_0$ und m-äquivalent.

1.12 Satz

Das **initiale Halteproblem** $H_{init} = e \in \mathbb{N}_0 = \Phi_e(0)\downarrow$ ist nicht entscheidbar.

Idee:

suche $f : \mathbb{N}_0 \to \mathbb{N}_0$ mit $\Phi_e(e)\downarrow \Leftrightarrow \Phi_{f(e)}(0)\downarrow$ Wähle f so dass $\Phi_{f(e)}(x) = \Phi_e(e)\ \forall x \in \mathbb{N}_0$

Beweis. Sei $\psi : \mathbb{N}_0^2 \rightsquigarrow \mathbb{N}_0$ mit $\psi(e,x) = \Phi_e(e) \forall e,x \in \mathbb{N}_0$. Dann ist ψ partiell berechenbar. Sei e_0 ein Index von ψ und $s : \mathbb{N}_0^2 \to \mathbb{N}_0$ gilt. Sei $f : \mathbb{N}_0 \to \mathbb{N}_0$ mit $f(e) = s(e_0,e) \forall e / in \mathbb{N}_0$. Dann ist f berechenbar. $\forall e \in \mathbb{N}_0$ gilt.

$$e \in H_{diag} \Leftrightarrow \Phi_e(e)\downarrow \Leftrightarrow \psi(e,0)\downarrow \Leftrightarrow \Phi_{e_0}(e,0)\downarrow \Leftrightarrow \Phi_s(e_0,e)(0)\downarrow \Leftrightarrow \Phi_{f(e)}(0)\downarrow \Leftrightarrow f(e) \in H_{init}$$

Es gilt also $H_{diag} \leq_m H_{init}$, da H_{diag} nicht entscheidbar ist, ist damit H_{init} nicht entscheidbar. □

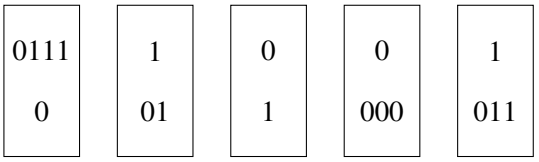
Dominosteinspiel!

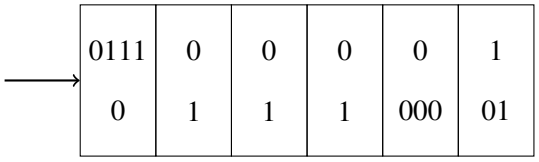
Gegeben:

Endlich viele typen von Spielsteinen mit jeweils zwei beschrifteten Feldern: öberes Feld, unteres Feld". Beschritungen sind nichtleere Wörter über einem Alphabet. Spielsteine sind vom gleichen Typ, wenn die beiden oberen Felder gleich beschriftet sind und die beiden unteren Felder gleich beschriftet sind. Es gibt von jedem Typ beliebig viele steine.

Gesucht:

Können ein oder mehrere (aber endlich viele) Spielsteine so nebeneinander gelegt werden, dass sich oben und unten von links nach rechts gelesen das gleiche Wort ergibt?





1.13 Definition (Postsches Korrespondenzproblem, Emil Port, 1946)

Für ein Alphabet Σ sei eine Instanz des Postschen Korrespondenzproblems über Σ eine endliche Teilmenge $I \subseteq (\Sigma^+)^2$. Eine Lösung für eine solche Instanz ist eine endliche Folge $(u_1, v_1), \cdots, (u_n, v_n)$ von Paaren in I mit $n \geq 1$,so dass

$$u_1 \cdots u_n = v_1 \cdots v_n$$

Gibt es eine Lösung für eine instanz des Postschen Korrespondenzproblems, so heißt diese Instanz lösbar. Das **Postsche Korrespondenzproblem** über einem Alphabet Σ, kurz PCP_Σ ist die Menge aller lösbaren Instanzen des Postschen Korrespondenzproblems über Σ.

Für ein Alphabet Σ sei eine Instanz des modifizierten Postschen Korrespondenzproblems über Σ ein Paar (p, I), wobei $I \subseteq (\Sigma^+)^2$ eine endliche Teilmenge und $p \in I$ ein Paar von Wörtern ist. Eine Lösung für eine solche Instanz ist eine endlcihe Folge $(u_1, v_1), \cdots, (u_n, v_n)$ von Paaren ist I, so dass

$$p = (u_1, v_1) und u_1 \cdots u_n = v_1 \cdots v_n$$

Gibt es eine Lösung für eine Instanz des modifizierten Postschen Korrespondenzproblems so heißt diese Instanz lösbar. Das **modifizierte Postsche Korrespondenzproblem** über einem Alphabet Σ, kurz $MPCP_\Sigma$ ist die Menge aller lösbaren Instanzen des modifizierten Postschen Korrespondenzproblems über Σ.

Plan:

Für Alphabet mit $|\Sigma| \geq 2$:

$$H_{init} \overset{(3)}{\leq_m} MPCP_\Gamma \overset{(2)}{\leq_m} PCP_\Gamma \overset{(1)}{\leq_m} PCP_\Sigma$$

1.14 Lemma

Für ein Alphabet Σ und Γ mit $|\Sigma| \geq w$ gilt $PCP_\Gamma \leq_m PCP_\Sigma$

Beweis. Wir suchen eine effektive Transformation, die jede Instanz I des Postschen Korrespondenzproblems über Γ in eine Instanz I' des postschen Korrespondenzproblems über Σ transformiert, so dass I genau dann lösbar ist, wenn I' lösbar ist. Seien $a_1, a_2 \in \Sigma$ verschieden und sein $b_1, \cdots, b_{|\Gamma|}$ die Elemente von Γ. Es bezeichne $\varphi : \Gamma^* \to \Gamma^*$ den eindeutigen Homomorphismus von Sprachen mit $\varphi(b_i) = a_1^i a_2$ $\forall i \in [|\Gamma|]$. Gegeben eine solche Instanz I wie oben sei $I' := \{(\varphi(u), \varphi(v)) : (u, v) \in I\}$. Die Funktion, die geeignete Codes von Instanzen I auf geeignete Codes von Instanzen I' abbildet ist berechenbar. Ist $(u_1, v_1), \cdots, (u_n, v_n)$ eine lösung I, so gilt

$$\varphi(u_1) \cdots \varphi(v_1) = \varphi(u_1, \cdots, \varphi(v_n)) = \varphi(v_1, \cdots, v_n) = \varphi(v_1) \cdots \varphi(v_n)$$

und somit ist $(\varphi(u_1), \varphi(v_1), \cdots, (\varphi(u_n)), \varphi(v_n))$ eine Lösung von I'. Die Instanz I' ist also lösbar wenn I lösbar ist. Ist $(u'_1, v'_1), \cdots, (u'_n, v'_n)$ eine Lösung von I', so gibt es eine Folge $(u_1, v_1) \cdots (u_n, v_n)$ von Paaren in I mit $\varphi(u'_i)$ und $\varphi(v_i) = v'_i$ $\forall i \in [n]$, also mit

$$\varphi(u_1, \cdots, u_n) = u'_1, \cdots, u'_n = u'_1, \cdots, u'_n = \varphi(u_1, \cdots, u_n)$$

Da $\varphi|_\Sigma$ injektiv und $\varphi(\Sigma)$ präfixfrei ist, ist φ injektiv (siehe Übung), folglich gilt $u_1, \cdots, u_n = v_1, \cdots, v_n$ und somit ist $(u_1, v_1), \cdots, (u_n, v_n)$ eine Lösung von I. Die Instanz I ist also lösbar wenn I' Lösbar ist. □

1.15 Lemma

Für Jedes alphabet Σ mit $|\Sigma| \leq w$ gitl $MPCP_\Sigma \leq_m PCP_\Sigma$.

Beweis. Sei Σ ein Alphabet mit $|\Sigma| \geq 2$. Nach lemma 3.14 genügt es ein Alphabet Γ zu finden, so das $MPCP_\Sigma \leq_m PCP_\Gamma$ gilt.
Wir suchen eine effektive Transformation , die jede instanz (p,I) des modifizierten Postschen Korrespondenzproblems über Σ in eine Instanz I' des Postschen Korrespondenzproblems über einem geeignetem AlphabetΓ transformiert, so dass (p,I) genau dann lösbar ist, wenn I' lösbar ist. □

Idee:

0	1	0	0	1	0	1	1	1	0	1
0	1	0	0	1	0	1	1	1	0	1

... Betrachte die Homomorphismus von Sprachen $\delta_\rightarrow$, $\delta_\leftarrow : \Sigma^* \rightarrow (\Sigma \cup *)^*$ mit $\delta_a = a*$ und $\delta_\leftarrow(a) = *a$ $\forall a \in \Sigma$. Für jede Instanz $(p,I) = ((u_1,v_1),I)$ wie oben sei

$$I' = \{(\delta_\leftarrow(u_1), *\delta_\rightarrow(v_1))\} \cup \{\delta_{leftarrow}(u), \delta_{rightarrow}(v) : (u,v) \in I\} \cup \{\delta_\leftarrow(u)*, \delta_\rightarrow(v) : (u,v) \in I\}$$

Die Funktion die geeignete Codes von Instanzen (p,I) auf geeignete Codes der zugehörigen Instanzen I' abbildet ist berechenbar. Gibt es eine Lösung $(u_1,v_1),\cdots,(u_n,u_n)$ von (p,I) dann ist

$$\begin{aligned}\delta_\leftarrow(u_1)\cdots\delta_\leftarrow(u_n)* &= \delta_\leftarrow(u_1\cdots u_n)* \\ &= \delta_\leftarrow(v_1\cdots v_n)* \\ &= *\delta_\rightarrow(v_1\cdots v_n) \\ &= *\delta_\rightarrow(v_1)\cdots\delta_\rightarrow(v_n)\end{aligned}$$

und folglich ist

$$(\delta_\leftarrow(u_1), *\delta_\rightarrow(v_1)), (\delta_\leftarrow(u_2), \delta_\rightarrow(v_2)), \cdots, (\delta_\leftarrow(u_{n-1}), \delta_\rightarrow(v_{n-1})), (\delta_\leftarrow(u_n), \delta_\rightarrow(v_n))$$

eine Lösung von I'. Es bleibt zu zeigen das (p,I) lösbar ist, wenn I' lösbar ist. Sei $\tau : (\Sigma \cup \{*\})^* \rightarrow \Sigma^*$ der Homomorphismus von Sprachen mit $\tau|_\Sigma$