Random reals and infinite time Turing machines

Philipp Schlicht, Universität Bonn
Merlin Carl, Universität Konstanz

10. August 2016

Berechenbarkeit aus vielen Orakeln

μ bezeichnet das Lebesgue-Mass auf den reellen Zahlen.

Lemma (Sacks, de Leuw-Moore-Shannon-Shapiro)

Angenommen A ist eine Menge reeller Zahlen mit $\mu(A) > 0$. Angenommen x ist Turing-berechenbar aus allen Elementen y von A. Dann ist x Turing-berechenbar.

Beweis.

Wir können annehmen, dass es ein einziges Programm gibt, das x aus allen Elementen von A berechnet.

Für $s : n \to \{0,1\}$ ist $I_s = \{x \mid \forall i < \text{length}(s) \ (i \in x \Leftrightarrow s(i) = 1)\}$.

Es gibt ein Intervall I_s, so dass $\frac{\mu(A \cap I_s)}{\mu(I_s)} > 0.5$ nach Lebesgue's density theorem. Wir können also annehmen, dass $\mu(A) > 0.5$.

Für jeden Wert $x(i)$ gibt es endlich viele $s_0, \ldots, s_n$ with $\mu(I_{s_0} \cup \ldots \cup I_{s_n}) > 0.5$, aus denen $x(i)$ berechnet wird.

Wir wählen die Mehrheitsentscheidung der $s_0, \ldots, s_n$. $\square$

Der Beweis funktioniert für ITTMs nicht, da unendliche Eingaben gelesen werden können.

Turing-Maschinen

Eine (deterministische) Turing-Maschine ist ein idealisierter Computer mit beliebig viel Speicher und beliebiger (endlicher) Laufzeit.

Eine (deterministische) Turing-Maschine besteht aus der folgenden Hardware

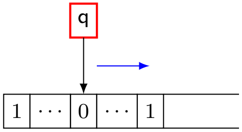
- Ein Band der Länge der natürlichen Zahlen ω
- Ein Lese-und Schreibkopf

und der folgenden Software

- Ein endliches Alphabet A, das ein Leerzeichen – enthält
- Eine endliche Menge von Zuständen, darunter einen Anfangszustand q_0 und Endzustände
- Eine Übergangsfunktion $\Delta\colon A \times S \to A \times S \times \{\text{links}, \text{rechts}\}$

Die Eingabe ist ein endliches Wort $w\colon n \to A$, das am Anfang auf dem Band steht.

Der Lese- und Schreibkopf der Turing-Maschine folgt der Übergangsfunktion.



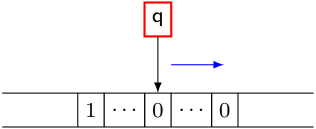
Turing-Maschinen

Der Maschine beginnt links auf dem Band (in Position 0).

Angenommen, zur Zeit t seht an der entsprechenden Stelle auf dem Band der Buchstabe a, und der aktuelle Zustand ist q.

Angenommen $T(a, q) = (b, r, \text{rechts})$.

- Ersetze a durch b.
- Wechsle vom Zustand q in den Zustand r.
- Bewege den Kopf ein Feld nach rechts.



Die Berechnung hält, sobald ein Endzustand erreicht ist. Andernfalls divergiert sie.

Die Ausgabe ist das Wort, das nach Ende der Berechnung auf dem Band steht (falls die Berechnung hält).

Unendliche Turing-Maschinen

Eine Infinite Time Turing Machine (ITTM, Hamkins-Kidder 2000) besteht aus der gleichen Hardware wie eine Turing-Maschine

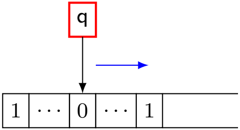
- Ein Band der Länge der natürlichen Zahlen ω
- Ein Lese-und Schreibkopf

und der folgenden Software

- Ein endliches Alphabet A, das ein Leerzeichen – enthält
- Eine endliche Menge von Zuständen, darunter einen Anfangszustand q_0 und Endzustände q_{ja}, q_{nein}
- Eine Übergangsfunktion
  $\Delta\colon A \times S \times \{\text{Nachfolger}, \text{Limes}\} \to A \times S \times \{\text{links}, \text{rechts}\}$

Die Eingabe ist ein unendliches Wort $w\colon \omega \to A$, das am Anfang auf dem Band steht.
Der Lese- und Schreibkopf der ITTM folgt der Übergangsfunktion.

Eine ITTM durchläuft nach allen endlichen Zeiten auch die unendlichen Ordinalzahlen ω, $\omega + 1$, $\omega + 2$,... $\omega \cdot 2$, $\omega \cdot 2 + 1$,... ω^2,... .



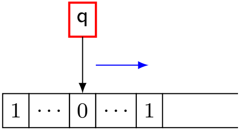
Unendliche Turing-Maschinen

Für jeden Schritt von Zeitpunkt α zum Zeitpunkt $\alpha + 1$ folgt die ITTM der Übergangsfunktion.

Die Buchstaben des Alphabets und die Zustände sind natürliche Zahlen.

Für jeden Limeszeitpunkt λ funktioniert die ITTM wie folgt.

- In jeder einzelnen Zelle des Bandes wird der liminf der vorherigen Zelleninhalte gebildet
- Der Zustand q_λ zur Zeit λ ist der liminf der früheren Zustände
- Der Lese- und Schreibkopf bewegt sich zum liminf n der vorherigen Positionen mit Zustand q_λ falls $n < \omega$, und zu Zelle 0 sonst.



Unendliche Berechnungen

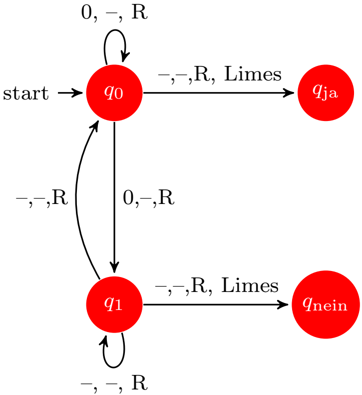
tape cells →

time ↓

time	state	head	0	1	2	3	4	5	⋯
0	0	0	–	–	–	–	–	–	⋯
1	1	1	1	–	–	–	–	–	⋯
2	0	2	1	–	–	–	–	–	⋯
3	1	3	1	–	1	–	–	–	⋯
4	0	4	1	–	1	–	–	–	⋯
5	1	5	1	–	1	–	1	–	⋯
6	0	6	1	–	1	–	1	–	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ω	0	0	1	–	1	–	1	–	⋯
$\omega + 1$	1	1	0	–	1	–	1	–	⋯
$\omega + 1$	0	2	0	–	1	–	1	–	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\omega \cdot 2$	0	0	0	–	0	–	0	–	⋯
$\omega \cdot 2 + 1$	1	1	1	–	0	–	0	–	⋯
$\omega \cdot 2 + 2$	0	2	1	–	0	–	0	–	⋯

Beispiel

Das Alphabet ist $A = \{0, -\}$. Kommt der Buchstabe 0 unendlich oft vor?



Beispiele

Beispiel

Das Halteproblem für Turing-Maschinen kann von einer ITTM berechnet werden.

Beispiel

Die Fundiertheit eines Baums $T \subseteq \omega^{<\omega}$, der durch eine binäre Relation $\dot{<}$ auf ω gegeben ist, kann durch eine ITTM berechnet werden.

ITTM-schreibbare reelle Zahlen

Definition

Angenommen x und y sind reelle Zahlen.

- x ist ITTM-schreibbar in y wenn es eine ITTM gibt, die mit Eingabe y die Ausgabe x hat.
- x ist ITTM-schliesslich schreibbar in y wenn es eine ITTM gibt, so dass mit Eingabe y der Inhalt des Bands gegen x konvergiert.
- x is ITTM- zufällig schreibbar in y wenn es eine ITTM gibt, so dass mit Eingabe y der Inhalt des Bands zu irgendeinem Zeitpunkt genau x ist.

Die Definition für OTMs ist analog.

Beispiel

Das Halteproblem für ITTMs ist ITTM-schliesslich schreibbar.

Definition

Eine Ordinalzahl α is schreibbar (schliesslich schreibbar, zufällig schreibbar) in y wenn es eine schreibbare (schliesslich schreibbare, zufällig schreibbare) zweistellige Relation $\dot{<}$ auf ω gibt, so dass $\langle \omega, \dot{<} \rangle$ isomorph zu $\langle \alpha, < \rangle$ ist.

ITTM-schreibbare Ordinalzahlen

Definition

Angenommen x ist eine reelle Zahl.

- λ^x ist das Supremum der ITTM-schreibbaren Ordinalzahlen in x.
- ζ^x ist das Supremum der ITTM-schliesslich schreibbaren Ordinalzahlen in x.
- Σ^x ist das Supremum der ITTM-zufällig schreibbaren Ordinalzahlen in x.

Eine Σ_n-Formel ist von der Form $\exists x_1 \forall x_2 ... \exists/\forall x_n \varphi(x_1, ..., x_n, y_1, ..., y_m)$, wobei die Quantoren in φ beschränkt sind.

Satz (Welch)

- *$\langle \lambda^x, \zeta^x, \Sigma^x \rangle$ ist das lexikographisch kleinste Tripel $\langle \alpha, \beta, \gamma \rangle$ von Ordinalzahlen $\alpha < \beta < \gamma$, so dass*
$$L_\alpha[x] \prec_{\Sigma_1} L_\beta[x] \prec_{\Sigma_2} L_\gamma[x]$$
- *Die schreibbaren (schliesslich schreibbaren, zufällig schreibbaren) reellen Zahlen in x sind genau die reellen Zahlen in $L_{\lambda^x}[x]$ ($L_{\zeta^x}[x]$, $L_{\Sigma^x}[x]$).*

Haltezeiten

Lemma (Hamkins-Lewis)

Das Supremum the Haltezeiten von ITTMs ist gleich dem Supremum λ der ITTM-schreibbaren Ordinalzahlen.

Berechenbarkeit aus vielen Orakeln

Wir schreiben $x \leqslant_s y$, $x \leqslant_{ss} y$, $y \leqslant_{zs} y$, wenn x schreibbar, schliesslich schreibbar oder zufällig schreibbar im Orakel y ist.

Theorem (CS)

Angenommen $x \subseteq \omega$.

1. x *ist ITTM-schreibbar genau dann, wenn* $\mu(\{y : x \leqslant_s y\}) > 0$
2. x*ist ITTM-schliesslich schreibbar genau dann, wenn* $\mu(\{y : x \leqslant_{ss} y\}) > 0$
3. x*ist ITTM-zufällig schreibbar genau dann, wenn* $\mu(\{y : x \leqslant_{zs} y\}) > 0$

Erkennbarkeit aus vielen Orakeln

Definition

Angenommen $Y \subseteq P(\omega)$ und $x \subseteq \omega$. x ist uniform $ITTM$-erkennbar aus Y genau dann, wenn es ein $ITTM$-Programm P gibt, so dass $P^z\downarrow$ für alle $z \subseteq \omega$ und $P^{z \oplus y} \downarrow = \delta_{z,x}$ für alle $y \in Y$.

Theorem (CS)

Angenommen $x \subseteq \omega$ ist uniform $ITTM$-erkennbar aus Y und $\mu(Y) > 0$. Dann ist x $ITTM$-erkennbar.

ITTM-Zufälligkeit

Definition

$x \subseteq \omega$ ist ITTM-zufällig, wenn es keine $ITTM$-semi-entscheidbare Menge X mit $x \in X$ und $\mu(X) = 0$ gibt.

Theorem (CS)

Angenommen $x, y \subseteq \omega$. $x \oplus y$ ist $ITTM$-zufällig genau dann, wenn x $ITTM$-zufällig ist und y $ITTM$-zufällig relativ zu x ist.