VL-08: Turing-Mächtigkeit

(Berechenbarkeit und Komplexität, WS 2018)

Gerhard Woeginger

WS 2018, RWTH

Organisatorisches

- Nächste Vorlesung:
  Donnerstag, November 29, 12:30–14:00 Uhr, Aula

- Webseite:
  http://algo.rwth-aachen.de/Lehre/WS1819/BuK.php

Wiederholung

Wdh.: Das Postsche Correspondenzproblem

Formale Definition (Postsches Correspondenzproblem, PCP)

Eine Instanz des PCP besteht aus einer endlichen Menge

$$K = \left\{ \left[\frac{x_1}{y_1}\right], \ldots, \left[\frac{x_k}{y_k}\right] \right\}$$

wobei $x_1, \ldots, x_k$ und $y_1, \ldots, y_k$ nichtleere Wörter über einem endlichen Alphabet Σ sind.

Das Problem besteht darin, zu entscheiden, ob es eine (nicht-leere) correspondierende Folge $\langle i_1, \ldots, i_n \rangle$ von Indizes in $\{1, \ldots, k\}$ gibt, sodass

$$x_{i_1} x_{i_2} \ldots x_{i_n} = y_{i_1} y_{i_2} \ldots y_{i_n}$$

Die Elemente von K nennen wir Dominosteine oder Dominos.

Wdh.: Unser Arbeitsplan

Satz A

$MPCP \leq PCP$

Satz B

$H \leq MPCP$

Die Transitivität der Reduzierbarkeit (Übung) impliziert $H \leq PCP$.

Satz

Das PCP ist unentscheidbar.

Vorlesung VL-08
Turing-Mächtigkeit

- Entscheidungsprobleme für CFGs
- Integration in geschlossener Form
- Satz von Richardson
- Hilberts zehntes Problem
- Satz von Matiyasevich

- Turing-Mächtigkeit

Entscheidungsprobleme für CFGs

Context-freie Grammatiken (1)

Aus der FOSAP Vorlesung:

Definition

Eine context-freie Grammatik (CFG) G ist ein Quadrupel (N, Σ, P, S), wobei

- N = Menge der Non-Terminal Symbole
- Σ = Terminal Alphabet
- P = Menge von Regeln der Form $A \to w$
  wobei $A \in N$ und $w \in (\Sigma \cup N)^*$
- S = Element von N (Startsymbol)

Also: Bei jeder Regel in P steht auf der linken Seite nur ein einzelnes Non-Terminal Symbol

Context-freie Grammatiken (2)

Beispiel

- $N = \{S\}$
- $\Sigma = \{a, b, c\}$
- P: $S \to aSa \,|\, bSb \,|\, cSc$; $S \to a \,|\, b \,|\, c$; $S \to aa \,|\, bb \,|\, cc$

Herleitung:

$S \to aSa \to acSca \to acaSaca \to acaaSaaca \to acaabbaaca$

Herleitung:

$S \to bSb \to bbSbb \to bbbSbbb \to bbbbbbbb$

Definition

$L(G)$ ist die Menge aller Worte über dem Terminal Alphabet Σ, die durch wiederholte Anwendung von Regeln in P aus dem Startsymbol S hergeleitet werden können.

Entscheidungsprobleme für CFGs

Die folgenden Probleme für CFGs sind entscheidbar:

- Gegeben CFG $\langle G \rangle$ und $w \in \Sigma^*$, gilt $w \in L(G)$?
- Gegeben CFG $\langle G \rangle$, ist $L(G)$ leer?
- Gegeben CFG $\langle G \rangle$, ist $L(G)$ endlich?

Die folgenden Probleme für CFGs sind unentscheidbar:

- Gegeben CFG $\langle G \rangle$, ist G eindeutig?
- Gegeben CFG $\langle G \rangle$, gilt $L(G) = \Sigma^*$?
- Gegeben CFG $\langle G \rangle$, ist $L(G)$ regulär?
- Gegeben CFGs $\langle G_1 \rangle$ und $\langle G_2 \rangle$, gilt $L(G_1) \subseteq L(G_2)$?
- Gegeben CFGs $\langle G_1 \rangle$ und $\langle G_2 \rangle$, ist $L(G_1) \cap L(G_2)$ leer?

Leerheit des Durchschnitts

Satz

Es ist unentscheidbar, ob die von zwei gegebenen CFGs G_1 und G_2 erzeugten Sprachen leeren Durchschnitt haben.

Beweisskizze:

- Betrachte eine beliebige PCP Instanz $\left\{ \left[\frac{x_1}{y_1}\right], \ldots, \left[\frac{x_k}{y_k}\right] \right\}$
- Es seien a, b, c drei Buchstaben, die nicht in x_i und y_i vorommen
- Konstruiere CFGs G_1 und G_2 mit folgenden Regeln:

$$G_1: \quad S \to x_1 S a^1 b \mid x_2 S a^2 b \mid x_3 S a^3 b \mid \cdots \mid x_k S a^k b \mid c$$

$$G_2: \quad S \to y_1 S a^1 b \mid y_2 S a^2 b \mid y_3 S a^3 b \mid \cdots \mid y_k S a^k b \mid c$$

- PCP lösbar genau dann wenn $L(G_1) \cap L(G_2)$ nicht leer

$$G_1: \quad S \overset{*}{\to} x_1\, x_4\, x_2\, x_5\, x_1\, x_4\ c\ a^4 b\, a^1 b\, a^5 b\, a^2 b\, a^4 b\, a^1 b$$

$$G_2: \quad S \overset{*}{\to} y_1\, y_4\, y_2\, y_5\, y_1\, y_4\ c\ a^4 b\, a^1 b\, a^5 b\, a^2 b\, a^4 b\, a^1 b$$

Integration in geschlossener Form

Integration in geschlossener Form (1)

Beispiel: Einige indefinite Integrale

$$\int 4x^3 + 3x^2 + 2x + 7 \, \mathrm{d}x = x^4 + x^3 + x^2 + 7x + C$$

$$\int x \sin(x) \, \mathrm{d}x = -x \cos(x) + \sin(x) + C$$

$$\int \frac{\sin(x)}{x} \, \mathrm{d}x = ???$$

$$\int e^{-x^2} \, \mathrm{d}x = ???$$

Integration in geschlossener Form (2)

Rationale Funktionen sind immer geschlossen integrierbar:

Satz (Liouville, 1838)

Wenn eine Funktion $f(x)$ der Quotient von zwei Polynomen $P(x)$ und $Q(x)$ ist, so kann $f(x)$ als Summe von Termen der Form

$$\frac{a}{(x-b)^n} \quad \text{und} \quad \frac{ax+b}{((x-c)^2+d^2)^n}$$

geschrieben werden.

Da jeder derartige Term geschlossen integrierbar ist,
ist jede rationale Funktion $f(x)$ geschlossen integrierbar.

Der Satz von Richardson

Eine Funktion heisst elementar, wenn sie durch Kombination von
Addition, Subtraktion, Multiplikation, Division,
Potenzieren, Wurzel ziehen, Logarithmieren, Betragsfunktion
und den trigonometrischen Funktionen konstruiert werden kann.

Im Jahr 1968 bewies der Britische Mathematiker Daniel Richardson das folgende Unentscheidbarkeitsresultat:

Satz von Richardson (1968)

Es ist unentscheidbar, ob eine gegebene elementare Funktion eine elementare Stammfunktion besitzt.

Der Beweis reduziert (im Wesentlichen) das Halteproblem für Turing Maschinen auf das Integrationsproblem.

Hilberts zehntes Problem

David Hilbert (1862–1943)

Wikipedia:
David Hilbert war ein deutscher Mathematiker. Er gilt als einer der bedeutendsten Mathematiker der Neuzeit. Viele seiner Arbeiten auf dem Gebiet der Mathematik und mathematischen Physik begründeten eigenständige Forschungsgebiete.

Hilbert begründete die moderne formalistische Auffassung von den Grundlagen der Mathematik und veranlasste eine kritische Analyse der mathematischen Begriffsdefinitionen und des mathematischen Beweises.

Hilberts zehntes Problem

Auf dem Internationalen Mathematikercongress in Paris im Jahr 1900 präsentierte David Hilbert eine Liste mit 23 mathematischen Problemen.

Hilberts zehntes Problem (im Originalwortlaut)

Eine *Diophantische* Gleichung mit irgend welchen Unbekannten und mit ganzen rationalen Zahlencoefficienten sei vorgelegt: man soll ein Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden lässt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.

- Mit den "ganzen rationalen Zahlen" sind einfach unsere ganzen Zahlen in $\mathbb{Z}$ gemeint.
- "Diophantische Gleichungen" sind Gleichungen mit Polynomen in einer Variablen (univariat) oder in mehreren Variablen (multivariat).

Diophantische Gleichungen

- Ein Term ist ein Produkt aus Variablen mit einem konstanten Koeffizienten. Zum Beispiel ist

  $$6 \cdot x \cdot x \cdot x \cdot y \cdot z \cdot z \qquad \text{bzw.} \qquad 6x^3yz^2$$

  ein Term über den Variablen x, y, z mit dem Koeffizienten 6.

- Ein Polynom ist eine endliche Summe von Termen, z.B.

  $$6x^3yz^2 + 3xy^2 - x^3 - 10$$

- Eine diophantische Gleichung setzt ein Polynom gleich Null. Die Lösungen der Gleichung entsprechen also den Nullstellen des Polynoms. Obiges Polynom führt zur diophantischen Gleichung $6x^3yz^2 + 3xy^2 - x^3 = 10$ mit der Nullstelle

  $$(x, y, z) = (5, 3, 0)$$

Beispiele (1)

Beispiel 29

Besitzt die Gleichung $x^3 + y^3 + z^3 = 29$ eine ganzzahlige Lösung?

Ja, zum Beispiel $(x, y, z) = (1, 1, 3)$

Beispiel 30

Besitzt die Gleichung $x^3 + y^3 + z^3 = 30$ eine ganzzahlige Lösung?

Ja, zum Beispiel $(x, y, z) = (-283059965, -2218888517, 2220422932)$
(Entdeckt von: Beck, Pine, Tarrant & Yarbrough, 2007)

Beispiele (2)

Beispiel 31

Besitzt die Gleichung $x^3 + y^3 + z^3 = 31$ eine ganzzahlige Lösung?

Nein!

- Modulo 9 genommen kann eine ganze Zahl n nur die Reste $0, \pm 1, \pm 2, \pm 3, \pm 4$ annehmen.
- Modulo 9 genommen kann eine Kubikzahl n^3 daher nur die Reste $0^3, (\pm 1)^3, (\pm 2)^3, (\pm 3)^3, (\pm 4)^3$ annehmen.
- Modulo 9 genommen kann eine Kubikzahl n^3 daher nur die Reste $0, \pm 1$ annehmen.
- Modulo 9 genommen kann die Summe $x^3 + y^3 + z^3$ von drei Kubikzahlen daher nur die Reste $0, \pm 1, \pm 2, \pm 3$ annehmen, und niemals die Reste ± 4.
- Da $31 \equiv 4 \pmod 9$, gibt es keine ganzzahlige Lösung.

Beispiele (3)

Beispiel 32

Besitzt die Gleichung $x^3 + y^3 + z^3 = 32$ eine ganzzahlige Lösung?

Nein! Da $32 \equiv -4 \pmod 9$, gibt es keine ganzzahlige Lösung.

Beispiel 33

Besitzt die Gleichung $x^3 + y^3 + z^3 = 33$ eine ganzzahlige Lösung?

Offenes Problem.
Man weiss nur: Es gibt keine Lösungen mit $|x|, |y|, |z| \leq 10^{12}$.

Beispiele (4)

Einige Turing-entscheidbare Beispiele

- Quadratische Gleichung $5x^2 - 3x + 6 = 0$.
  Lösbar mit Mittelschul-Methoden. Daher kann man auch feststellen, ob die Gleichung ganzzahlige Lösungen hat.

- Diophantische Gleichungen in einer Variablen x:

  $$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 = 0$$

  Jede ganzzahlige Lösung x ist ein Teiler von a_0.
  Man muss also nur die Werte x mit $|x| \leq |a_0|$ durchprobieren.

- Pell'sche Gleichung $x^2 = c\,y^2 + 1$ mit positiv ganzzahligem c.
  Wenn c keine Quadratzahl ist, so besitzt die Pell'sche Gleichung unendlich viele positive Lösungen (x, y).

Formulierung als Entscheidungsproblem

Hilberts zehntes Problem (im Originalwortlaut)

Eine *Diophantische* Gleichung mit irgend welchen Unbekannten und mit ganzen rationalen Zahlencoefficienten sei vorgelegt: man soll ein Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden lässt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.

Hilberts zehntes Problem (in moderner Formulierung)

Konstruiere einen Algorithmus, der entscheidet, ob ein gegebenes Polynom mit ganzzahligen Koeffizienten eine ganzzahlige Nullstelle hat.

Hilberts zehntes Problem (in BuK Formulierung)

Konstruiere eine Turing Maschine, die die folgende Sprache entscheidet:

$$\text{Dioph} = \{ \langle p \rangle \mid p \text{ ist ein Polynom mit ganzzahligen Koeffizienten und mit (mindestens) einer ganzzahligen Nullstelle} \}$$

Rekursive Aufzählbarkeit von Dioph

Für ein Polynom $p \in \mathbb{Z}[x_1, \ldots, x_\ell]$ in ℓ Variablen
entspricht der Wertebereich der abzählbar unendlichen Menge $\mathbb{Z}^\ell$.

Der folgende Algorithmus erkennt daher Dioph:

- Zähle die ℓ-Tupel aus $\mathbb{Z}^\ell$ in kanonischer Reihenfolge auf
  und werte p für jedes dieser Tupel aus.
- Akzeptiere, sobald eine dieser Auswertungen den Wert 0 ergibt.

Wir folgern:

Satz

Die Sprache Dioph ist rekursiv aufzählbar.

Entscheidbarkeit von Dioph (1)

- Falls wir eine obere Schranke für die Absolutwerte der Nullstellen hätten, so brauchten wir nur eine endliche Menge von ℓ-Tupeln aufzuzählen. Damit wäre das Problem Dioph entscheidbar.
- Wir haben bereits gesehen:
  Für univariate Polynome $p(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$ ist $|a_0|$ eine derartige obere Schranke.
- Für multivariate Polynome gibt es aber keine derartige obere Schranke für die Absolutwerte der Nullstellen: Betrachten Sie zum Beispiel das Polynom $p(x, y) = x + y$.
- Andrerseits: Wir brauchen ja gar keine Schranke, die für alle Nullstellen gilt. Es reicht schon, wenn eine einzige der Nullstellen beschränkt ist.
- Existiert eine derartige Schranke? Oder gibt es vielleicht ganz andere Möglichkeiten, um einem Polynom anzusehen, ob es eine ganzzahlige Nullstelle besitzt?

Entscheidbarkeit von Dioph (2)

Erst siebzig Jahre, nachdem Hilbert sein Problem präsentiert hatte, beantwortete der russische Mathematiker Yuri Matiyasevich diese Fragen:

Satz von Matiyasevich (1970)

Das Problem, ob ein ganzzahliges Polynom eine ganzzahlige Nullstelle besitzt, ist unentscheidbar.

- Der Beweis beruht auf einer langen Kette von Reduktionen, durch die letztendlich das Halteproblem für Turing Maschinen auf das Nullstellenproblem Dioph reduziert wird.
- Yuri Matiyasevich hat das letzte Glied in dieser Kette geschlossen.
- Andere wichtige Kettenglieder wurden in den Jahren 1950–1970 von Martin Davis, Julia Robinson und Hilary Putnam konstruiert.

Entscheidbarkeit von Dioph (3)

Tatsächlich folgt aus dieser Beweiskette ein stärkeres Resultat:

Satz (Davis, Robinson, Putnam & Matiyasevich)

Für jede Teilmenge $X \subseteq \mathbb{Z}$ der ganzen Zahlen sind die beiden folgenden Aussagen äquivalent:

- X ist rekursiv aufzählbar
- Es existiert ein $(k+1)$-variates Polynom $p(x, y_1, \ldots, y_k)$ mit ganzzahligen Koeffizienten, sodass
  $X = \{x \in \mathbb{Z} \mid \exists y_1, \ldots, y_k \in \mathbb{Z} \text{ mit } p(x, y_1, \ldots, y_k) = 0\}$

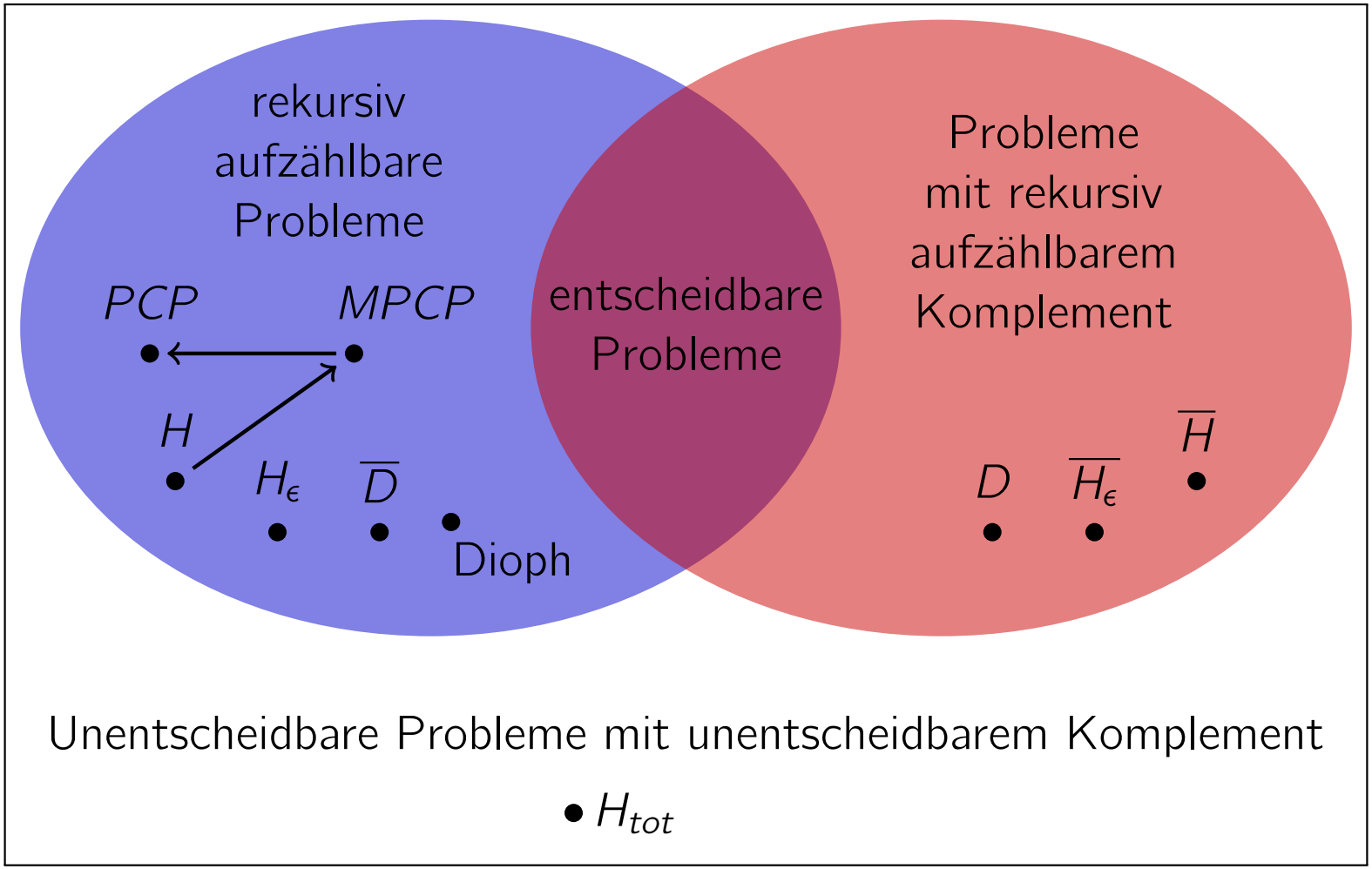
- Ganzzahlige Polynome sind also genauso "berechnungsstark" und genauso "mächtig" wie Turing Maschinen.
- Der Beweis ist technisch und lang (...viel, viel, viel zu lang, um in dieser Vorlesung präsentiert zu werden).

Beispiel: Ein Primzahl-generierendes Polynom

Man erhält genau die Menge der Primzahlen, wenn man die 26 Variablen im folgenden Polynom durch 26 beliebige nicht-negative ganze Zahlen ersetzt und alle resultierenden nicht-negativen Werte ausdruckt:

$$
\begin{aligned}
(k+2)\{1-(&[wz+h+j-q]^2 \\
&+ [(gk+2g+k+1)(h+j)+h-z]^2 \\
&+ [16(k+1)^3(k+2)(n+1)^2+1-f^2]^2 \\
&+ [2n+p+q+z-e]^2+[e^3(e+2)(a+1)^2+1-o^2]^2 \\
&+ [(a^2-1)y^2+1-x^2]^2+[16r^2y^4(a^2-1)+1-u^2]^2 \\
&+ [((a+u^2(u^2-a))2-1)(n+4dy)^2+1-(x+cu)^2]^2 \\
&+ [(a^2-1)\ell^2+1-m^2]^2 \\
&+ [ai+k+1-\ell-i]^2+[n+\ell+v-y]^2 \\
&+ [p+\ell(a-n-1)+b(2an+2a-n^2-2n-2)-m]^2 \\
&+ [q+y(a-p-1)+s(2ap+2a-p^2-2p-2)-x]^2 \\
&+ [z+p\ell(a-p)+t(2ap-p^2-1)-pm]^2)\}
\end{aligned}
$$

Berechenbarkeitslandschaft



Turing-Mächtigkeit

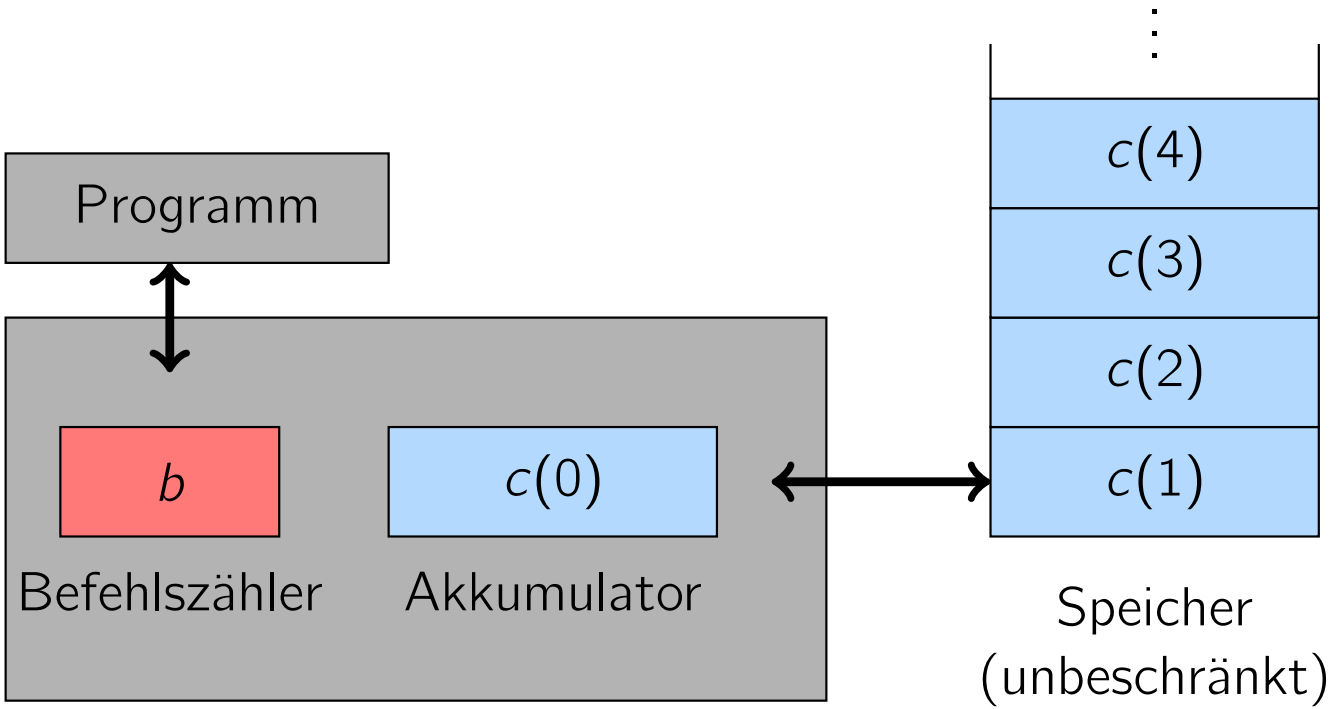
Turing-mächtige Rechnermodelle

Definition

Ein Rechnermodell wird als Turing-mächtig bezeichnet,
wenn jede Funktion, die durch eine TM berechnet werden kann,
auch durch dieses Rechnermodell berechnet werden kann.

- Da die Registermaschine (RAM) die Turingmaschine simulieren kann, ist sie Turing-mächtig
- Auch eine viel schwächere Variante der RAM mit stark eingeschränktem Befehlssatz ist Turing-mächtig

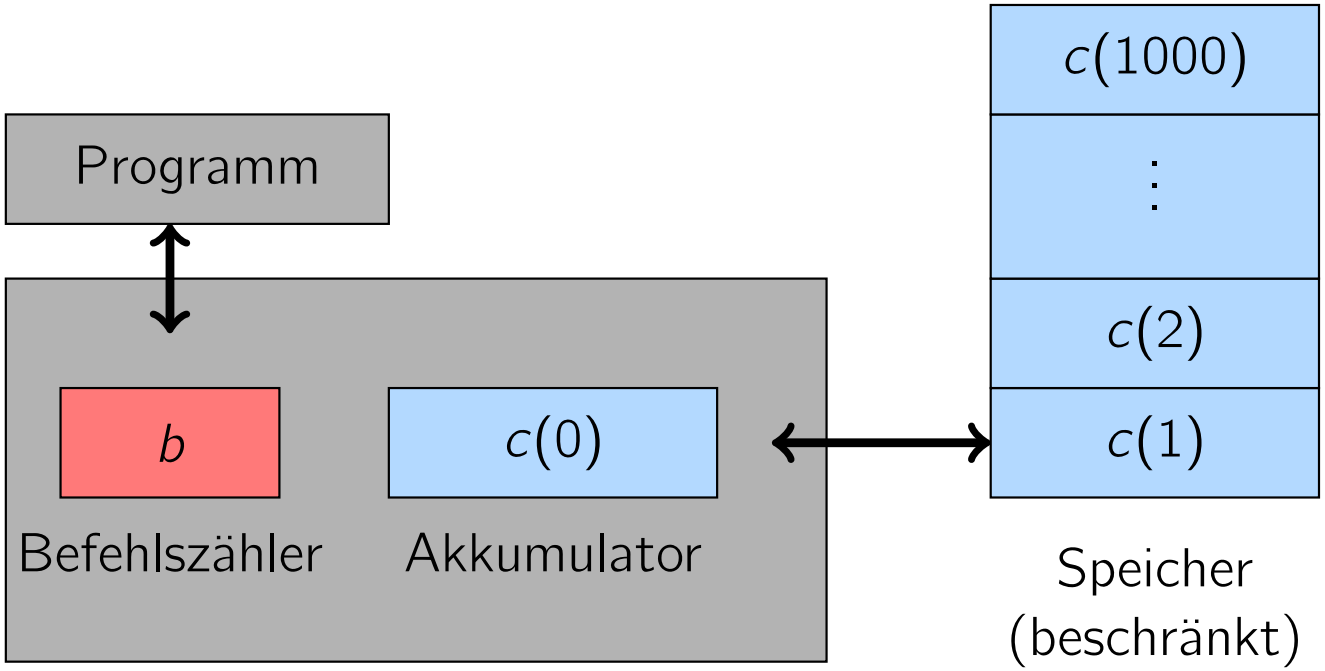
Registermaschinen (RAM)



Befehlssatz:

LOAD, STORE, ADD, SUB, MULT, DIV
INDLOAD, INDSTORE, INDADD, INDSUB, INDMULT, INDDIV
CLOAD, CADD, CSUB, CMULT, CDIV
GOTO, IF c(0)?x THEN GOTO (mit ? aus {=,<,<=,>,>=})
END

Eingeschränkte Registermaschine (Mini-RAM)



Eingeschränkter Befehlssatz:

LOAD, STORE, ~~ADD~~, ~~SUB~~, ~~MULT~~, ~~DIV~~
~~INDLOAD~~, ~~INDSTORE~~, ~~INDADD~~, ~~INDSUB~~, ~~INDMULT~~, ~~INDDIV~~
CLOAD, CADD, CSUB, ~~CMULT~~, ~~CDIV~~
GOTO, IF c(0)>0 THEN GOTO ~~(mit ? aus {=,<,<=,>,>=})~~
END

Mächtigkeit der Mini-RAM

Übung

Die Mini-RAM ist eine eingeschränkte Variante der RAM

- mit nur acht Befehlen LOAD, STORE, CLOAD, CADD, CSUB, GOTO, IF c(0)>0 THEN GOTO, END
- und mit einer endlichen Anzahl von Registern.

Zeigen Sie, dass die Mini-RAM Turing-mächtig ist.

Hinweis: Diese Übung wird viel einfacher, wenn man erst einmal den Stoff von VL-09 gesehen hat.

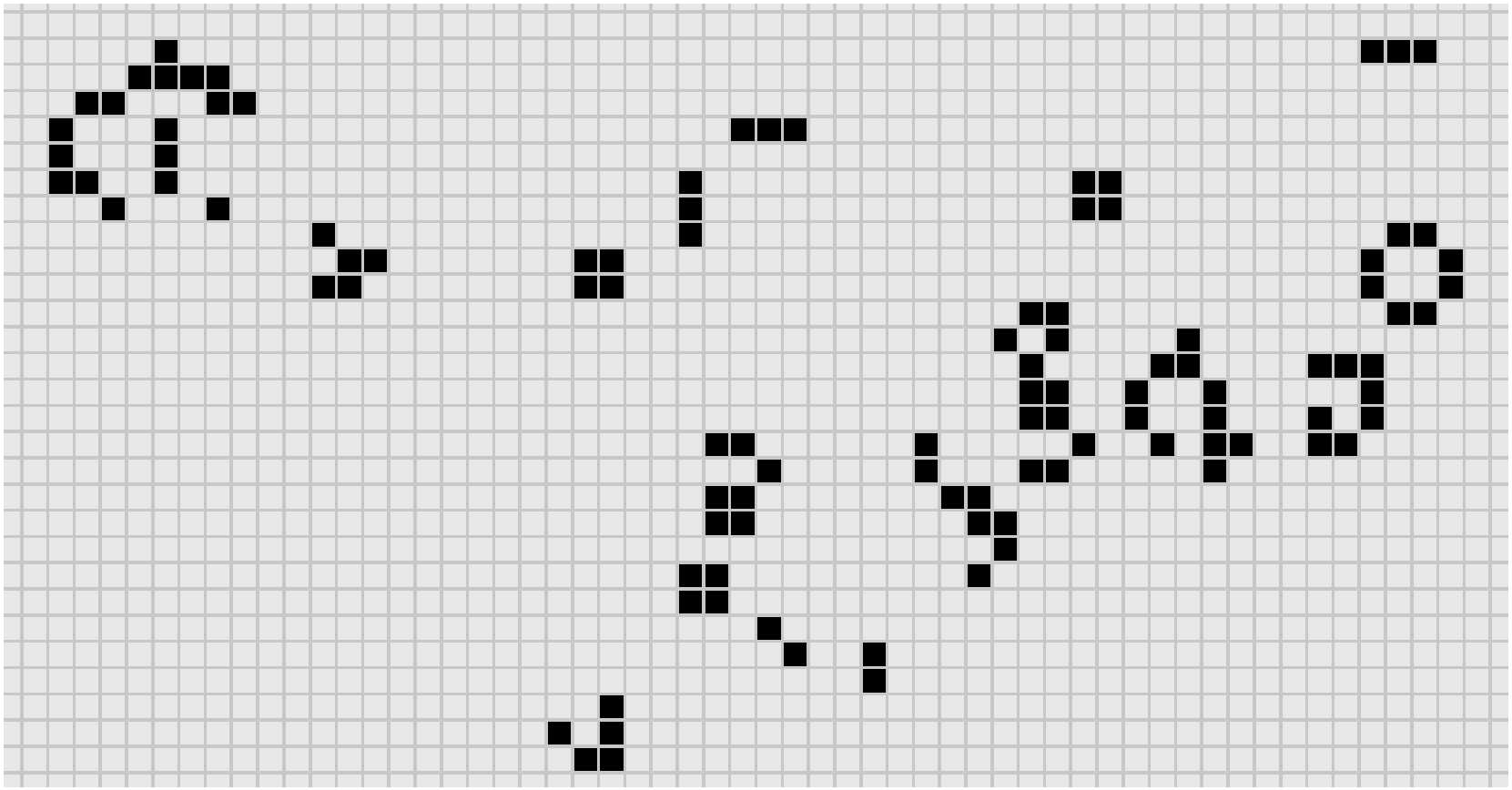
Turing-Mächtigkeit: Beispiele

- Reines HTML (ohne JavaScript; ohne Browser) ist nicht Turing-mächtig
- Tabellenkalkulationen (ohne Schleifen) sind nicht Turing-mächtig

- Der Lambda Calculus von Alonzo Church ist äquivalent zur TM, und daher Turing-mächtig
- Die μ-rekursiven Funktionen von Stephen Kleene sind äquivalent zur TM, und daher Turing-mächtig
- Alle gängigen höheren Programmiersprachen sind Turing-mächtig: Algol, Pascal, C, FORTRAN, COBOL, Java, Smalltalk, Ada, C++, Python, LISP, Haskell, PROLOG, etc.
- PostScript, Tex, Latex sind Turing-mächtig
- Sogar PowerPoint ist Turing-mächtig (wegen Animated Features)

Conway's Game of Life

Turing-Mächtigkeit: Game of Life

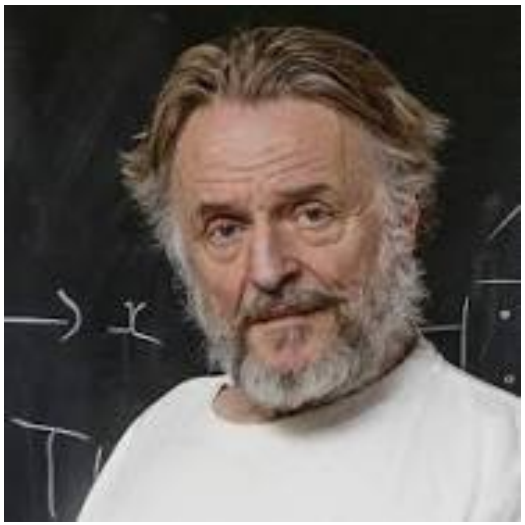


John Horton Conway's "Game of Life" ist Turing-mächtig

John Horton Conway FRS (1937)

Wikipedia: John Horton Conway is an English mathematician active in the theory of finite groups, knot theory, number theory, combinatorial game theory, and coding theory. He has also contributed to many branches of recreational mathematics, and invented the Game of Life.

In 2004, Conway and Simon Kochen proved the free will theorem, a startling version of the no-hidden-variables principle of quantum mechanics. It states that (given certain conditions) if an experimenter can freely decide what quantities to measure in a particular experiment, then elementary particles must be free to choose their spins to make the measurements consistent with physical law.

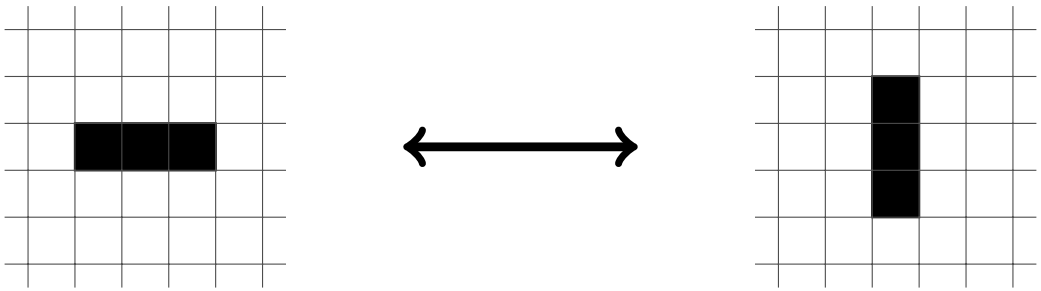


Game of Life: Regeln

Conway's Game of Life (aus dem Jahr 1970) ist ein zellulärer Automat, der auf einem unendlichen 2-dimensionalen Gitter arbeitet.
Zu jedem Zeitpunkt ist jede Zelle entweder lebend oder tot.

Ausgehend von einer Anfangskonfiguration entwickelt sich die Zellenkonfiguration Schritt für Schritt folgendermassen:

- Eine tote Zelle mit genau drei lebenden Nachbarn ist im nächsten Schritt lebendig.
- Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben ab.
- Lebende Zellen mit mehr als drei lebenden Nachbarn sterben ab.
- Alle anderen Zellen bleiben unverändert.



Game of Life: Fragen

Kombinatorische Fragen

- Kann eine Anfangskonfiguration K unbeschränkt wachsen?
- Gibt es für jede Anfangskonfiguration K eine Schranke $f(K)$, sodass alle aus K entstehenden Konfigurationen höchstens $f(K)$ lebende Zellen haben?

Algorithmische Fragen

Falls das Game of Life mit einer gegebenen Anfangskonfiguration K gestartet wird,

- stirbt das System dann irgendwann aus?
- erreicht das System jemals einen stabilen Zustand?
- verhält sich das System dann irgendwann periodisch?
- erreicht das System dann jemals eine gegebene Zielkonfiguration K'?

Game of Life: Antworten

Satz

Conway's Game of Life ist Turing-mächtig.

Die folgenden Probleme sind unentscheidbar:

- Stirbt das System mit Anfangskonfiguration K jemals aus?
- Stabilisiert sich das System mit Anfangskonfiguration K jemals?
- Verhält sich das System mit Anfangskonfiguration K irgendwann einmal periodisch?
- Erreicht das System mit Anfangskonfiguration K jemals die Zielkonfiguration K'?

Ausserdem gibt es Anfangskonfigurationen, die unbeschränkt wachsen.