Script generated by TTT

Title: Seidl: Theoretische_Informatik (27.06.2013)

Date: Thu Jun 27 16:02:15 CEST 2013

Duration: 87:52 min

Pages: 97

Ackermann als Familie von Funktionen A_m: $A_m(n) := a(m,n)$.
Nun gilt:

$$\begin{aligned} A_0(n) &= s(n) \\ A_{m+1}(n) &= A_m^{n+1}(1) = \underbrace{A_m(\ldots A_m}_{n+1}(1)\ldots) \end{aligned}$$

Beobachtung: alle A_m sind total und PR (mit Ind. über m)

Mit Currying (z.B. in Haskell):

```
a 0      n  =  n+1
a (m+1)  n  =  iter (n+1) (a m) 1

iter 0      f x  =
iter (n+1)  f x  =
```


Ackermann als Familie von Funktionen A_m: $A_m(n) := a(m,n)$.
Nun gilt:

$$\begin{aligned} A_0(n) &= s(n) \\ A_{m+1}(n) &= A_m^{n+1}(1) = \underbrace{A_m(\ldots A_m}_{n+1}(1)\ldots) \end{aligned}$$

Beobachtung: alle A_m sind total und PR (mit Ind. über m)

Mit Currying (z.B. in Haskell):

```
a 0      n  =  n+1
a (m+1)  n  =  iter (n+1) (a m) 1
```


4.8 Die Ackermann-Funktion

$$\begin{aligned} a(\;\;0, \;\;n) &= n+1 \\ a(m+1, \;\;0) &= a(m,1) \\ a(m+1, n+1) &= a(m, a(m+1,n)) \end{aligned}$$

- Dies ist keine PR Definition.
- Aber: $\not\Rightarrow$ a ist nicht PR
- Ziel: a ist berechenbar, total, aber nicht PR

Fakt 4.47

Die Ackermann-Funktion ist (OCaml-)berechenbar.

0   iter n f (f x)

Ackermann als Familie von Funktionen A_m: $A_m(n) := a(m, n)$.
Nun gilt:

$$\begin{aligned} A_0(n) &= s(n) \\ A_{m+1}(n) &= A_m^{n+1}(1) = \underbrace{A_m(\ldots A_m}_{n+1}(1)\ldots) \end{aligned}$$

Beobachtung: alle A_m sind total und PR (mit Ind. über m)

Mit Currying (z.B. in Haskell):

```
a 0       n  =  n+1
a (m+1)   n  =  iter (n+1) (a m) 1

iter 0       f x  =
iter (n+1)   f x  =
```

Ackermann ist mit primitiver Rekursion höherer Stufe definierbar.

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m, n) \neq \perp$ (1)

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m, n) \neq \perp$ (1)

- Basis: $a(0, n) = n + 1 \neq \perp$

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m, n) \neq \bot$ (1)

- Basis: $a(0, n) = n + 1 \neq \bot$
- Schritt: $a(m + 1, n) \neq \bot$ (2)

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m, n) \neq \bot$ (1)

- Basis: $a(0, n) = n + 1 \neq \bot$
- Schritt: $a(m + 1, n) \neq \bot$ (2)

  Beweis mit Induktion über n:

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m, n) \neq \bot$ (1)

- Basis: $a(0, n) = n + 1 \neq \bot$
- Schritt: $a(m + 1, n) \neq \bot$ (2)

  Beweis mit Induktion über n:

  - $a(m + 1, 0) = a(m, 1) \neq \bot$ wg (1)

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m, n) \neq \bot$ (1)

- Basis: $a(0, n) = n + 1 \neq \bot$
- Schritt: $a(m + 1, n) \neq \bot$ (2)

  Beweis mit Induktion über n:

  - $a(m + 1, 0) = a(m, 1) \neq \bot$ wg (1)
  - $a(m + 1, n + 1)$

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m,n) \neq \perp \quad (1)$

- Basis: $a(0,n) = n + 1 \neq \perp$
- Schritt: $a(m+1,n) \neq \perp \quad (2)$

  Beweis mit Induktion über n:
  - $a(m+1,0) = a(m,1) \neq \perp \quad \text{wg } (1)$
  - $a(m+1,n+1) = a(m,a(m+1,n))$

Man kann auch direkt zeigen:

Lemma 4.48
Die Ackermann-Funktion ist total.

Beweis:
Mit Induktion über m: $\forall n \in \mathbb{N}.\ a(m,n) \neq \perp \quad (1)$

- Basis: $a(0,n) = n + 1 \neq \perp$
- Schritt: $a(m+1,n) \neq \perp \quad (2)$

  Beweis mit Induktion über n:
  - $a(m+1,0) = a(m,1) \neq \perp \quad \text{wg } (1)$
  - $a(m+1,n+1) = a(m,a(m+1,n)) = a(m,k)$

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. $\square$

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.
Details: [Felscher. *Berechenbarkeit*]

Satz 4.50
Die Ackermann-Funktion ist nicht PR.

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.
Details: [Felscher. *Berechenbarkeit*]

Satz 4.50
Die Ackermann-Funktion ist nicht PR.

Beweis:
Indirekt. Angenommen a wäre doch PR.

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.
Details: [Felscher. *Berechenbarkeit*]

Satz 4.50
Die Ackermann-Funktion ist nicht PR.

Beweis:
Indirekt. Angenommen a wäre doch PR. Dann ist auch f PR:

$$f(n) \ := \ a(n, n)$$

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.
Details: [Felscher. *Berechenbarkeit*]

Satz 4.50
Die Ackermann-Funktion ist nicht PR.

Beweis:
Indirekt. Angenommen a wäre doch PR. Dann ist auch f PR:

$$f(n) \ := \ a(n, n)$$

Nach obigem Lemma gibt es t mit $f(n) < a(t, n)$ für alle n.

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.
Details: [Felscher. *Berechenbarkeit*]

Satz 4.50
Die Ackermann-Funktion ist nicht PR.

Beweis:
Indirekt. Angenommen a wäre doch PR. Dann ist auch f PR:

$$f(n) \ := \ a(n, n)$$

Nach obigem Lemma gibt es t mit $f(n) < a(t, n)$ für alle n.

$$f(t) < a(t, t) = f(t)$$

Lemma 4.49
Für jede PR Funktion $f : \mathbb{N}^k \to \mathbb{N}$ gibt es ein $t \in \mathbb{N}$, so dass

$$\forall \overline{x} \in \mathbb{N}^k.\ f(\overline{x}) < a(t, \max \overline{x}).$$

Beweis:
Mit Induktion über den Aufbau der Definition von f. □

Prinzip: $t \approx$ Länge der Definition von f.
Details: [Felscher. *Berechenbarkeit*]

Satz 4.50
Die Ackermann-Funktion ist nicht PR.

Beweis:
Indirekt. Angenommen a wäre doch PR. Dann ist auch f PR:

$$f(n) \ := \ a(n, n)$$

Nach obigem Lemma gibt es t mit $f(n) < a(t, n)$ für alle n.

$$f(t) < a(t, t) = f(t) \qquad ↯$$

□

Oberflächlich intuitiv:

> *Die Ackermann-Funktion wächst schneller als alle PR Funktionen.*

Oberflächlich intuitiv:

Die Ackermann-Funktion wächst schneller als alle PR Funktionen.

Genauer:

Die Funktion $n \mapsto a(n, n)$ *wächst schneller als alle PR Funktionen.*

Oberflächlich intuitiv:

Die Ackermann-Funktion wächst schneller als alle PR Funktionen.

Genauer:

Die Funktion $n \mapsto a(n, n)$ *wächst schneller als alle PR Funktionen.*

Intuitiver Grund:

- Für fixes t ist $n \mapsto a(t, n)$ PR, denn dies ist A_t.

Oberflächlich intuitiv:

Die Ackermann-Funktion wächst schneller als alle PR Funktionen.

Genauer:

Die Funktion $n \mapsto a(n, n)$ *wächst schneller als alle PR Funktionen.*

Intuitiver Grund:

- Für fixes t ist $n \mapsto a(t, n)$ PR, denn dies ist A_t.
- A_t braucht aber eine PR Definition der Länge $O(t)$.

Da die Ackermann-Funktion total, berechenbar und nicht PR ist:

Korollar 4.51
Die PR Funktionen sind eine echte Teilklasse der berechenbaren totalen Funktionen.

Die berechenbaren Funktionen

berechenbar $=$ TM $=$ WHILE $=$ GOTO $= \mu$R

total (zB Ackermann)

LOOP $=$ PR

4.9 Entscheidbarkeit und das Halteproblem

4.9 Entscheidbarkeit und das Halteproblem

Ziel: Es ist unentscheidbar ob ein Programm terminiert.

4.9 Entscheidbarkeit und das Halteproblem

Ziel: Es ist unentscheidbar ob ein Programm terminiert.

Definition 4.52

Eine Menge A ($\subseteq \mathbb{N}$ oder Σ^*) heißt entscheidbar gdw ihre charakteristische Funktion

$$\chi_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{falls } x \notin A \end{cases}$$

berechenbar ist.

4.9 Entscheidbarkeit und das Halteproblem

Ziel: Es ist unentscheidbar ob ein Programm terminiert.

Definition 4.52

Eine Menge A ($\subseteq \mathbb{N}$ oder Σ^*) heißt entscheidbar gdw ihre charakteristische Funktion

$$\chi_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{falls } x \notin A \end{cases}$$

berechenbar ist.

Eine Eigenschaft/Problem $P(x)$ heißt entscheidbar gdw $\{x \mid P(x)\}$ entscheidbar ist.

4.9 Entscheidbarkeit und das Halteproblem

Ziel: Es ist unentscheidbar ob ein Programm terminiert.

Definition 4.52

Eine Menge A ($\subseteq \mathbb{N}$ oder Σ^*) heißt entscheidbar gdw ihre charakteristische Funktion

$$\chi_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{falls } x \notin A \end{cases}$$

berechenbar ist.

Eine Eigenschaft/Problem $P(x)$ heißt entscheidbar gdw $\{x \mid P(x)\}$ entscheidbar ist.

Fakt 4.53

Die entscheidbaren Mengen sind abgeschlossen unter Komplement: Ist A entscheidbar, dann auch $\overline{A}$.

Kodierung einer TM als Wort über $\Sigma = \{0, 1\}$, exemplarisch:

- Sei $\Gamma = \{a_0, \dots, a_k\}$ und $Q = \{q_0, \dots, q_n\}$.

Kodierung einer TM als Wort über $\Sigma = \{0, 1\}$, exemplarisch:

- Sei $\Gamma = \{a_0, \dots, a_k\}$ und $Q = \{q_0, \dots, q_n\}$.
- $\delta(q_i, a_j) = (q_{i'}, a_{j'}, d)$ wird kodiert als

$$\#bin(i)\#bin(j)\#bin(i')\#bin(j')\#bin(m)$$

  wobei $bin : \mathbb{N} \to \{0, 1\}^*$ die Binärkodierung einer Zahl ist und $m = 0/1/2$ falls $d = L/R/N$.

Kodierung einer TM als Wort über $\Sigma = \{0, 1\}$, exemplarisch:

- Sei $\Gamma = \{a_0, \dots, a_k\}$ und $Q = \{q_0, \dots, q_n\}$.
- $\delta(q_i, a_j) = (q_{i'}, a_{j'}, d)$ wird kodiert als

  $$\#bin(i)\#bin(j)\#bin(i')\#bin(j')\#bin(m)$$

  wobei $bin : \mathbb{N} \to \{0, 1\}^*$ die Binärkodierung einer Zahl ist und $m = 0/1/2$ falls $d = L/R/N$.
- Kodierung von δ: Konkatenation der Kodierungen aller $\delta(.,.) = (.,.,.)$, in beliebiger Reihenfolge.

Kodierung einer TM als Wort über $\Sigma = \{0, 1\}$, exemplarisch:

- Sei $\Gamma = \{a_0, \dots, a_k\}$ und $Q = \{q_0, \dots, q_n\}$.
- $\delta(q_i, a_j) = (q_{i'}, a_{j'}, d)$ wird kodiert als

  $$\#bin(i)\#bin(j)\#bin(i')\#bin(j')\#bin(m)$$

  wobei $bin : \mathbb{N} \to \{0, 1\}^*$ die Binärkodierung einer Zahl ist und $m = 0/1/2$ falls $d = L/R/N$.
- Kodierung von δ: Konkatenation der Kodierungen aller $\delta(.,.) = (.,.,.)$, in beliebiger Reihenfolge.
- Kodierung von $\{0, 1, \#\}^*$ in $\{0, 1\}^*$:

  $$\begin{array}{ccc} 0 & \mapsto & 00 \\ 1 & \mapsto & 01 \\ \# & \mapsto & 11 \end{array}$$

Die Kodierung TM $\to \{0, 1\}^*$ ist nicht surjektiv,

Kodierung einer TM als Wort über $\Sigma = \{0, 1\}$, exemplarisch:

- Sei $\Gamma = \{a_0, \dots, a_k\}$ und $Q = \{q_0, \dots, q_n\}$.
- $\delta(q_i, a_j) = (q_{i'}, a_{j'}, d)$ wird kodiert als

  $$\#bin(i)\#bin(j)\#bin(i')\#bin(j')\#bin(m)$$

  wobei $bin : \mathbb{N} \to \{0, 1\}^*$ die Binärkodierung einer Zahl ist und $m = 0/1/2$ falls $d = L/R/N$.
- Kodierung von δ: Konkatenation der Kodierungen aller $\delta(.,.) = (.,.,.)$, in beliebiger Reihenfolge.
- Kodierung von $\{0, 1, \#\}^*$ in $\{0, 1\}^*$:

  $$\begin{array}{ccc} 0 & \mapsto & 00 \\ 1 & \mapsto & 01 \\ \# & \mapsto & 11 \end{array}$$

Die Kodierung TM $\to \{0,1\}^*$ ist nicht surjektiv,

Die Kodierung TM $\to \{0,1\}^*$ ist nicht surjektiv,
dh nicht jedes Wort über $\{0,1\}^*$ kodiert eine TM.
Sei $\hat{M}$ eine beliebige feste TM.

Die Kodierung TM $\to \{0,1\}^*$ ist nicht surjektiv,
dh nicht jedes Wort über $\{0,1\}^*$ kodiert eine TM.
Sei $\hat{M}$ eine beliebige feste TM.

Definition 4.54
Die zu einem Wort $w \in \{0,1\}^*$ gehörige TM M_w ist

$$M_w := \begin{cases} M & \text{falls } w \text{ Kodierung von } M \text{ ist} \\ \hat{M} & \text{sonst} \end{cases}$$

Die Kodierung TM $\to \{0,1\}^*$ ist nicht surjektiv,
dh nicht jedes Wort über $\{0,1\}^*$ kodiert eine TM.
Sei $\hat{M}$ eine beliebige feste TM.

Definition 4.54
Die zu einem Wort $w \in \{0,1\}^*$ gehörige TM M_w ist

$$M_w := \begin{cases} M & \text{falls } w \text{ Kodierung von } M \text{ ist} \\ \hat{M} & \text{sonst} \end{cases}$$

Die Kodierung von syntaktischen Objekten (Programmen, Formeln, etc) als Zahlen nennt man Gödelisierung und die Zahlen Gödelnummern.

Definition 4.55

$M[w]$ ist Abk. für „Maschine M mit Eingabe w“

Definition 4.55

$M[w]$ ist Abk. für „Maschine M mit Eingabe w“

$M[w]\downarrow$ bedeutet, dass $M[w]$ terminiert/hält.

Definition 4.55

$M[w]$ ist Abk. für „Maschine M mit Eingabe w“

$M[w]\downarrow$ bedeutet, dass $M[w]$ terminiert/hält.

Definition 4.56 (Spezielles Halteproblem)

Gegeben: Ein Wort $w \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe w ?

Definition 4.55

$M[w]$ ist Abk. für „Maschine M mit Eingabe w“

$M[w]\downarrow$ bedeutet, dass $M[w]$ terminiert/hält.

Definition 4.56 (Spezielles Halteproblem)

Gegeben: Ein Wort $w \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe w ?

Als Menge:

$$K := \{w \in \{0,1\}^* \mid M_w[w]\downarrow\}$$

Satz 4.57

Das spezielle Halteproblem ist nicht entscheidbar.

Satz 4.57

Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.

Satz 4.57

Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \perp & \text{falls } \chi_K(w) = 1 \end{cases}$$

Satz 4.57

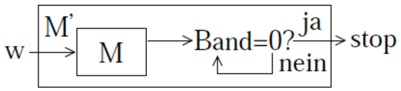
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \perp & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Satz 4.57

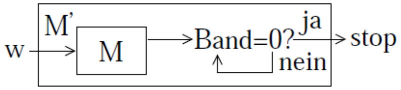
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \bot & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Dann gibt es ein w' mit $M_{w'} = M'$.

Satz 4.57

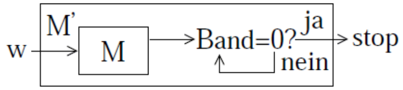
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \bot & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Dann gibt es ein w' mit $M_{w'} = M'$.

$$f(w') = \bot \quad \Leftrightarrow \quad \chi_K(w') = 1$$

Satz 4.57

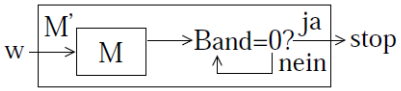
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \bot & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Dann gibt es ein w' mit $M_{w'} = M'$.

$$\begin{aligned} f(w') = \bot \quad &\Leftrightarrow \quad \chi_K(w') = 1 \\ &\Leftrightarrow \quad M_{w'}[w']{\downarrow} \end{aligned}$$

Satz 4.57

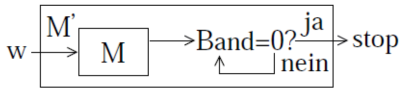
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \bot & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Dann gibt es ein w' mit $M_{w'} = M'$.

$$\begin{aligned} f(w') = \bot \quad &\Leftrightarrow \quad \chi_K(w') = 1 \\ &\Leftrightarrow \quad M_{w'}[w']{\downarrow} \end{aligned}$$

Satz 4.57

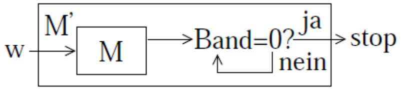
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \perp & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Dann gibt es ein w' mit $M_{w'} = M'$.

$$\begin{aligned} f(w') = \perp &\Leftrightarrow \chi_K(w') = 1 \\ &\Leftrightarrow M_{w'}[w']\downarrow \quad \Leftrightarrow \quad M'[w']\downarrow \end{aligned}$$

Satz 4.57

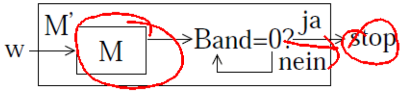
Das spezielle Halteproblem ist nicht entscheidbar.

Beweis:

Angenommen, K sei entscheidbar, dh χ_K ist berechenbar.
Dann ist auch f berechenbar:

$$f(w) := \begin{cases} 1 & \text{falls } \chi_K(w) = 0 \\ \perp & \text{falls } \chi_K(w) = 1 \end{cases}$$

Berechnet die TM M χ_K so berechnet M' f:



Dann gibt es ein w' mit $M_{w'} = M'$.

$$\begin{aligned} f(w') = \perp &\Leftrightarrow \chi_K(w') = 1 \\ &\Leftrightarrow M_{w'}[w']\downarrow \quad \Leftrightarrow \quad M'[w']\downarrow \quad \Leftrightarrow \quad f(w') = 1 \end{aligned}$$

Definition 4.58 ((Allgemeines) Halteproblem)

Gegeben: Wörter $w, x \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe x ?

Definition 4.58 ((Allgemeines) Halteproblem)

Gegeben: Wörter $w, x \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe x ?

Als Menge:

$$H := \{w\#x \mid M_w[x]\downarrow\}$$

Definition 4.58 ((Allgemeines) Halteproblem)

Gegeben: Wörter $w, x \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe x ?

Als Menge:

$$H := \{w\#x \mid M_w[x]\downarrow\}$$

Satz 4.59

Das Halteproblem H ist nicht entscheidbar.

Definition 4.58 ((Allgemeines) Halteproblem)

Gegeben: Wörter $w, x \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe x ?

Als Menge:

$$H := \{w\#x \mid M_w[x]\downarrow\}$$

Satz 4.59

Das Halteproblem H ist nicht entscheidbar.

Beweis:

Wäre H entscheidbar, dann trivialerweise auch K:

$$\chi_K(w) = \chi_H(w, w)$$

$\square$

Definition 4.58 ((Allgemeines) Halteproblem)

Gegeben: Wörter $w, x \in \{0,1\}^*$.

Problem: Hält M_w bei Eingabe x ?

Als Menge:

$$H := \{w\#x \mid M_w[x]\downarrow\}$$

Satz 4.59

Das Halteproblem H ist nicht entscheidbar.

Beweis:

Wäre H entscheidbar, dann trivialerweise auch K:

$$\chi_K(w) = \chi_H(w, w)$$

$\square$

Definition 4.60 (Reduktion)

Eine Menge $A \subseteq \Sigma^*$ ist reduzierbar auf eine Menge $B \subseteq \Gamma^*$

Definition 4.60 (Reduktion)

Eine Menge $A \subseteq \Sigma^*$ ist reduzierbar auf eine Menge $B \subseteq \Gamma^*$ gdw es eine totale und berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ gibt mit

$$\forall w \in \Sigma^*.\ w \in A \Leftrightarrow f(w) \in B$$

Definition 4.60 (Reduktion)

Eine Menge $A \subseteq \Sigma^*$ ist reduzierbar auf eine Menge $B \subseteq \Gamma^*$ gdw es eine totale und berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ gibt mit

$$\forall w \in \Sigma^*.\ w \in A \Leftrightarrow f(w) \in B$$

Wir schreiben dann $A \leq B$.

Definition 4.60 (Reduktion)

Eine Menge $A \subseteq \Sigma^*$ ist reduzierbar auf eine Menge $B \subseteq \Gamma^*$ gdw es eine totale und berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ gibt mit

$$\forall w \in \Sigma^*.\ w \in A \Leftrightarrow f(w) \in B$$

Wir schreiben dann $A \leq B$.

Intuition:

- B ist mindestens so schwer zu lösen wie A.

Definition 4.60 (Reduktion)

Eine Menge $A \subseteq \Sigma^*$ ist reduzierbar auf eine Menge $B \subseteq \Gamma^*$ gdw es eine totale und berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ gibt mit

$$\forall w \in \Sigma^*.\ w \in A \Leftrightarrow f(w) \in B$$

Wir schreiben dann $A \leq B$.

Intuition:

- B ist mindestens so schwer zu lösen wie A.
- Ist A unlösbar, dann auch B.

Definition 4.60 (Reduktion)

Eine Menge $A \subseteq \Sigma^*$ ist reduzierbar auf eine Menge $B \subseteq \Gamma^*$ gdw es eine totale und berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ gibt mit

$$\forall w \in \Sigma^*.\ w \in A \Leftrightarrow f(w) \in B$$

Wir schreiben dann $A \leq B$.

Intuition:

- B ist mindestens so schwer zu lösen wie A.
- Ist A unlösbar, dann auch B.
- Ist B lösbar, dann erst recht A.

Lemma 4.61

Falls $A \leq B$ und B ist entscheidbar, so ist auch A entscheidbar.

Lemma 4.61

Falls $A \leq B$ und B ist entscheidbar, so ist auch A entscheidbar.

Beweis:

Es gelte $A \leq B$ mittels f und χ_B sei berechenbar.

Lemma 4.61

Falls $A \leq B$ und B ist entscheidbar, so ist auch A entscheidbar.

Beweis:

Es gelte $A \leq B$ mittels f und χ_B sei berechenbar.

Dann ist $\chi_B \circ f$ berechenbar und $\chi_A = \chi_B \circ f$:

Lemma 4.61
Falls $A \leq B$ und B ist entscheidbar, so ist auch A entscheidbar.

Beweis:
Es gelte $A \leq B$ mittels f und χ_B sei berechenbar.
Dann ist $\chi_B \circ f$ berechenbar und $\chi_A = \chi_B \circ f$:

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} = \begin{cases} 1, & f(x) \in B \\ 0, & f(x) \notin B \end{cases} = \chi_B(f(x)) \qquad \square$$

Korollar 4.62
Falls $A \leq B$ und A ist unentscheidbar, dann ist auch B unentscheidbar.

Lemma 4.61
Falls $A \leq B$ und B ist entscheidbar, so ist auch A entscheidbar.

Beweis:
Es gelte $A \leq B$ mittels f und χ_B sei berechenbar.
Dann ist $\chi_B \circ f$ berechenbar und $\chi_A = \chi_B \circ f$:

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} = \begin{cases} 1, & f(x) \in B \\ 0, & f(x) \notin B \end{cases} = \chi_B(f(x)) \qquad \square$$

Korollar 4.62
Falls $A \leq B$ und A ist unentscheidbar, dann ist auch B unentscheidbar.

Beispiel 4.63
Da $K \leq H$ (mit Reduktion $f(w) := w\#w$) und K unentscheidbar ist, ist auch H unentscheidbar.

Satz 4.64
Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Satz 4.64
Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:
Wir zeigen $K \leq H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \leq H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \leq H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Überschreibe die Eingabe mit w; führe M_w aus.

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \leq H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Überschreibe die Eingabe mit w; führe M_w aus.

Dh f berechnet aus w die Kodierung w_1 einer TM, die w schreibt,

$f : w \mapsto w'$

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \leq H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Überschreibe die Eingabe mit w; führe M_w aus.

Dh f berechnet aus w die Kodierung w_1 einer TM, die w schreibt,

und gibt die Kodierung von "$w_1; w$" zurück.
Damit ist f total und berechenbar.

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \le H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Überschreibe die Eingabe mit w; führe M_w aus.

Dh f berechnet aus w die Kodierung w_1 einer TM, die w schreibt, und gibt die Kodierung von "$w_1; w$" zurück.
Damit ist f total und berechenbar.
Es gilt:

$$w \in K \;\Leftrightarrow\; M_w[w]\downarrow$$

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \le H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Überschreibe die Eingabe mit w; führe M_w aus.

Dh f berechnet aus w die Kodierung w_1 einer TM, die w schreibt, und gibt die Kodierung von "$w_1; w$" zurück.
Damit ist f total und berechenbar.
Es gilt:

$$w \in K \;\Leftrightarrow\; M_w[w]\downarrow \;\Leftrightarrow\; M_{f(w)}[\epsilon]\downarrow$$

Satz 4.64

Das Halteproblem auf leerem Band, H_0, ist unentscheidbar.

$$H_0 := \{w \in \{0,1\}^* \mid M_w[\epsilon]\downarrow\}$$

Beweis:

Wir zeigen $K \le H_0$ mit einer Funktion $f : \{0,1\}^* \to \{0,1\}^*$.
$f(w)$ ist die Gödelnummer folgender TM:

Überschreibe die Eingabe mit w; führe M_w aus.

Dh f berechnet aus w die Kodierung w_1 einer TM, die w schreibt, und gibt die Kodierung von "$w_1; w$" zurück.
Damit ist f total und berechenbar.
Es gilt:

$$w \in K \;\Leftrightarrow\; M_w[w]\downarrow \;\Leftrightarrow\; M_{f(w)}[\epsilon]\downarrow \;\Leftrightarrow\; f(w) \in H_0$$

□

Fazit:

Es gibt keine allgemeine algorithmische Methode, um zu entscheiden, ob ein Programm terminiert.

Fazit:
Es gibt keine allgemeine algorithmische Methode, um zu entscheiden, ob ein Programm terminiert.

Die Unentscheidbarkeit vieler Fragen über die Ausführung von Programmen folgt durch Reduktion des Halteproblems:

- Kann ein WHILE-Programm mit einer bestimmten Eingabe einen bestimmten Programmpunkt erreichen?

Fazit:
Es gibt keine allgemeine algorithmische Methode, um zu entscheiden, ob ein Programm terminiert.

Die Unentscheidbarkeit vieler Fragen über die Ausführung von Programmen folgt durch Reduktion des Halteproblems:

- Kann ein WHILE-Programm mit einer bestimmten Eingabe einen bestimmten Programmpunkt erreichen?
  Der Spezialfall Programmpunkt=Programmende ist das Halteproblem.

Fazit:
Es gibt keine allgemeine algorithmische Methode, um zu entscheiden, ob ein Programm terminiert.

Die Unentscheidbarkeit vieler Fragen über die Ausführung von Programmen folgt durch Reduktion des Halteproblems:

- Kann ein WHILE-Programm mit einer bestimmten Eingabe einen bestimmten Programmpunkt erreichen?
  Der Spezialfall Programmpunkt=Programmende ist das Halteproblem.
- Kann Variable x_7 bei einer bestimmten Eingabe je den Wert 2^{32} erreichen?

Quiz

Ist es entscheidbar, ob eine TM

1. mehr als 314 Zustände hat?

Quiz

Ist es entscheidbar, ob eine TM

2. bei Eingabe ϵ mehr als 314 Schritte macht?

Quiz

Ist es entscheidbar, ob eine TM

3. bei *irgendeiner* Eingabe mehr als 314 Schritte macht?

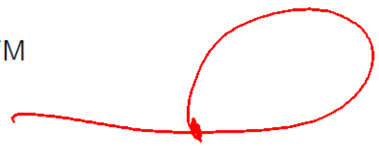
Quiz

Ist es entscheidbar, ob eine TM

4. bei *allen* Eingaben mehr als 314 Schritte macht?

Quiz

Ist es entscheidbar, ob eine TM



5. bei Eingabe ϵ ihren Kopf mehr als 314 Felder von der 0-Position entfernen kann?

Quiz

Ist es entscheidbar, ob eine TM

5. bei Eingabe ϵ ihren Kopf mehr als 314 Felder von der 0-Position entfernen kann?

Quiz

Ist es entscheidbar, ob eine TM

6. bei Eingabe ϵ einen bestimmten Zustand erreichen kann?

Quiz

Ist es entscheidbar, ob eine TM

7. *irgendeine* Eingabe akzeptieren kann?

Quiz

Ist es entscheidbar, ob eine TM

6. bei Eingabe ϵ einen bestimmten Zustand erreichen kann?

Quiz

Ist es entscheidbar, ob eine TM

- *irgendeine* Eingabe akzeptieren kann?

Quiz

Ist es entscheidbar, ob eine TM

- *irgendeine* Eingabe akzeptieren kann?