Theoretische Informatik

= mathematische Grundlagen der maschinellen Berechnung

- formale Sprachen und Automaten
- Berechenbarkeitstheorie
- Komplexitätstheorie

---

Computer gibt es in vielen Spielarten

- Desktop oder Laptop
- Smartphones
- eingebettete Systeme
- Supercomputer
- Relaiscomputer
- Quantencomputer
- DNA-Computer

→ Gemeinsamkeit:
Alle Arten von Computern können etwas berechnen!

→ Funktionen berechnen!
Gegeben $f: \mathbb{N} \to \mathbb{N}$.
Man möchte für $n$ $f(n)$ berechnen.

# Welche Funktionen sind berechenbar?

1. Versuch:

**Def.** Eine (möglicherweise partielle) Funktion $f: \mathbb{N} \to \mathbb{N}$ heißt *berechenbar* (im intuitiven Sinne), falls es eine *Rechenvorschrift* gibt, die für jeden Wert im Definitionsbereich von $f$ das Ergebnis $f(n)$ anzeigt.

Rechenvorschrift:
- Algorithmus
- Programm
- $\mu$-rekursive Funktion
- Turing-Maschine

~~Touring~~

Turing-Maschinen führen Operationen auf Strings aus, nehmen Eingabe entgegen, Ausgabe.

→ TMs sind mit endlichen Mitteln beschreibbar

2. Versuch:

Def Eine Funktion $f: |X| \to \mathbb{N}$ heißt Turing-berechenbar, falls es eine Turing-Maschine (TM) gibt, die für den Definitionsbereich von f angesetzt auf die Eingabe $n$, $f(n)$ auf das Band schreibt.

Man stellt fest: TMs können alle Arten von anderen Rechenmaschinen simulieren und umgekehrt!

# Church-Turing-These

Die Menge aller im intuitiven Sinne berechenbaren Funktionen ist die Menge der Turing-berechenbaren Funktionen.

Gibt es nicht-berechenbare Fkt.?

**Def.** Eine Menge $M$ heißt abzählbar, falls es eine surjektive Fkt. $f : \mathbb{N} \to M$ gibt.

**Bsp:** $\mathbb{N}$ ist per Def. abzählbar

Primzahlen sind abzählbar

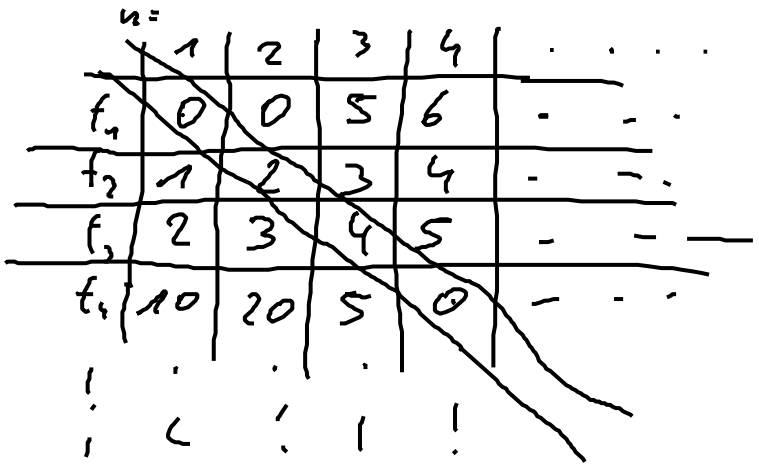
| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | ... |
|---|---|---|---|---|---|---|---|
| $f(n)$ | 2 | 3 | 5 | 7 | 11 | 13 | ... |

**Satz** Die Menge der Turing-berechenbaren Fkt. ist abzählbar.

**Bew:** Die Menge der TMs ist abzählbar. Diese Abzählung gibt uns alle Turing-berechenbare Fkt. □

**Satz 2** Es existieren Funktionen, die nicht Turing-berechenbar sind.

**Bew:** Sei $f_1, f_2, \ldots$ eine Abzählung der TM-berechenbaren Fkt.




$$f_d(i) = \begin{cases} 0 & \text{falls } f_i(i) \neq 0 \\ 1 & \text{falls } f_i(i) = 0 \end{cases}$$

$f_d(1) = 1$

$f_d(2) = 0$

$f_d(3) = 0$

$f_d(4) = 1$

$\vdots$

D.h. es ex. kein $i$ mit $f_i = f_d$!

$f_d$ kommt in der Liste nicht vor!

D.h. $f_d$ ist nicht berechenbar. ⊏

Bsp: Sicherheitskritisches System

→ Sie wollen ein Tool haben, das überprüft, dass eine von Ihnen programmierte Funktion eine bestimmte Eigenschaft (z.B. Totschleifenfreiheit).

→ So ein Tool ist leider unmöglich.

Auswege:

1. Eingeschränkte Formalismen zur Spezifikation der sicherheitskrit. Teile einsetzen.
2. Unterapproximationen berechnen.

Allgemein: Probleme im Kontext von Verifikation und Synthese von Hard- und Softwaresystemen sind nicht algorithmisch lösbar.

**Satz** (Pumping-Lemma)

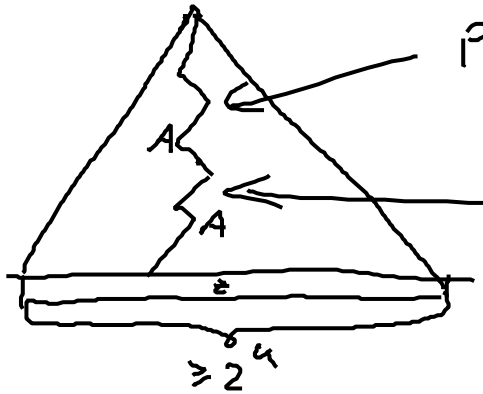
Sei $L$ eine kontextfreie Sprache. Dann gibt es $n \in \mathbb{N}$, so dass sich alle Wörter $z \in L$ und $|z| \geq n$ zerlegen lassen in $z = uvwxy$ und:

(1) $|vx| \geq 1$

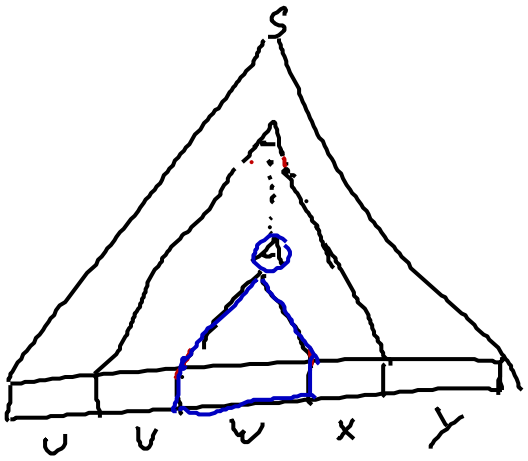
(2) $|vwx| \leq n$

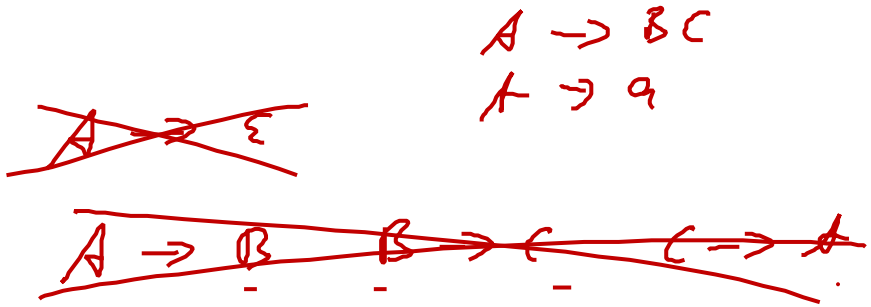
(3) $uv^i w x^i y \in L \quad \forall i \geq 0$

**Beweis:** Sei $G = (V, \Sigma, P, S)$ eine Grammatik für $L - \{\varepsilon\}$ in CNF. Sei $k = |V|$. Wähle $n = 2^k$ (wenn es solche Länge nicht gibt, dann sind wir fertig). Betrachte Syntaxbaum für bel. $z \in L$ und $|z| \geq n$.




Pfad der Länge $k \Rightarrow$ mind. $k+1$ Var-Symbole und mind. 1 Var-Symbol kommt 2x vor.



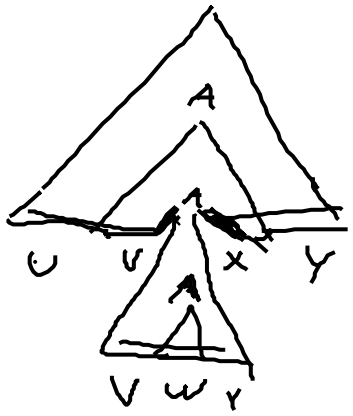




(1) $|vx| \geq 1$. Da die Gram. in CNF ist, muss entweder v oder x (oder beide) nicht-leer sein.

(2) $|vwx| \leq k$. Wir suchen von unten nach oben nach dem ersten Doppelvorkommen. Dann ist oberes A max. k Schritte von dem unteren entfernt. Vom oberen A sind es max. k Schritte zu den Blättern. Unterhalb des oberen A's gibt es max $2^k = n$ Blätter.

(3) $uv^i wx^i y \in L$.


$u v^0 w x^0 y$




$u v^2 w x^2 y$

- - - - - -

D.h. (3) gilt auch □

---

$L = \{ a^k b^m c^k d^m \mid k, m \geq 1 \}$

Bsp: $L$ ist nicht kontextfrei.

Wir nehmen an, dass $L$ kf ist. D.h. es ex.
$n$ mit der PL - Eigenschaft. Wähle $z = a^n b^n c^n d^n$.
D.h. $|z| = 4n$. $|vx| \geq 1$ und $|vwx| \leq n$. Damit gilt
dass $vwx$ besteht aus (1) nur aus a's (2) aus a's und b's
(3) aus b's, (4) aus b's und c's, (5) c's, (6) aus c's und d's

und (7) muss aus d's. Da $uv^n wx^n y \in L$ nach (3), folgt also, dass entweder unterschiedliche Anzahlen von b's und c's oder von b's und d's in w sind. Widerspruch zur Def. der Sprache. □