

Formale Sprachen

ELIZA

Turing - Test

Berechenbarkeit

Turing Maschine

Formale Sprachen

Struktur durch Aufgabe „Kommunikation mit einer Maschine“ geprägt

Formale Grammatik $G = (N, \Sigma, P, S)$

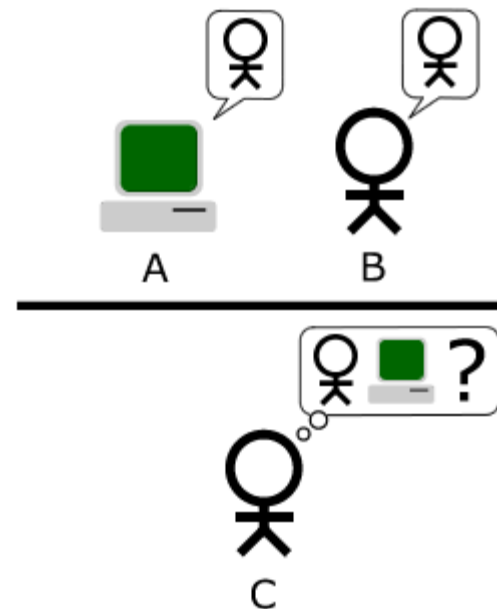
Kommunikation mit Maschine über natürliche Sprache

- Beispiel: ELIZA

(Joseph Weizenbaum, 1966)

Turing Test

- 1950 von Alan Turing (1912 – 1954) zur Feststellung künstlicher Intelligenz vorgeschlagen



Quelle: <http://de.wikipedia.org/wiki/Turing-Test>

Kritik am Turing Test

- Testet nur Unterhaltung nicht Intelligenz
- Maschine auch ohne Kommunikationsfähigkeit intelligent ?!
- Menschen bestehen möglicherweise auch nicht
- Nicht standardisierbar
- Prüft Funktionalität nicht Bewusstsein (J. Searle -> Chinesisches Zimmer)

Kritik am Turing Test

- Turing: im Jahr 2000 höchstens 70% Chance für Anwender Mensch/Maschine zu identifizieren
- zu optimistisch, natürliche Intelligenz zu komplex?!
- Cleverbot (2011): ca. 60% von 1334 Personen für Mensch gehalten (www.cleverbot.com)
- Philip K. Dick: *Do Androids Dream of Electric Sheep?* (↑ *Blade Runner* von R. Scott) → Test auf Empathie

Formale Sprachen

Wie lassen sich formale Sprachen beschreiben?

1. Aufzählung aller Wörter $L = \{aa, ab, ba, bb\}$
2. Spezifikation eines formalen Ausdrucks
 $L = \{a, b\}^2$
3. Durch Grammatiken $L(G)$ mit $G = (N, \Sigma, P, S)$
4. Mit Automaten $L(A)$ $A = (Q, \Sigma, \delta, q_0, F)$

Formale Sprachen - Automaten

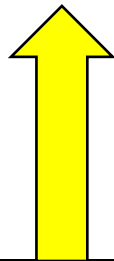
Grammatik		Sprachen	Entscheidbarkeit	Automaten
Typ 0	beliebige Formale Grammatik	rekursiv aufzählbar		Turingmaschine
Typ 1	Kontextsensitive Grammatik	kontextsensitiv	Wortproblem	linear beschränkter Automat
Typ 2	Kontextfreie Grammatik	kontextfrei	Wortproblem Leerheitsproblem Endlichkeitsproblem	nicht deterministischer Kellerautomat
Typ 3	Reguläre Grammatik	regulär	Wortproblem Leerheitsproblem Endlichkeitsproblem Äquivalenzproblem	endlicher Automat

Turingmaschine (A. Turing, 1936)

Band



Schreib-/Lesekopf



Arbeitsschritte		Zustand		q ₀
Zustand	Symbol	Zustand	Schreibe	gehe nach
q ₀	1	q ₀	1	R
q ₀	0	q ₁	1	R
q ₁	0	q ₁	1	R
q ₁	1	q _H	1	R

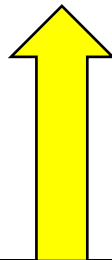
Kontrolleinheit

Turingmaschine

Band



Schreib-/Lesekopf



Arbeitsschritte		Zustand		q ₀
Zustand	Symbol	Zustand	Schreibe	gehe nach
q ₀	1	q ₀	1	R
q ₀	0	q ₁	1	R
q ₁	0	q ₁	1	R
q ₁	1	q _H	1	R

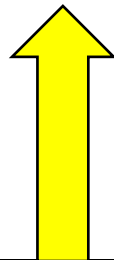
Kontrolleinheit

Turingmaschine

Band



Schreib-/Lesekopf



Arbeitsschritte		Zustand		q ₁
Zustand	Symbol	Zustand	Schreibe	gehe nach
q ₀	1	q ₀	1	R
q ₀	0	q ₁	1	R
q ₁	0	q ₁	1	R
q ₁	1	q _H	1	R

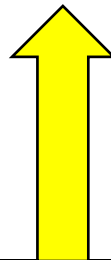
Kontrolleinheit

Turingmaschine

Band



Schreib-/Lesekopf



Arbeitsschritte		Zustand		
Zustand	Symbol	Zustand	Schreibe	gehe nach
q_0	1	q_0	1	R
q_0	0	q_1	1	R
q_1	0	q_1	1	R
q_1	1	q_H	1	R

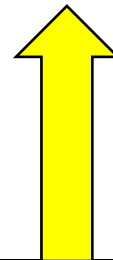
Kontrolleinheit

Turingmaschine

Band



Schreib-/Lesekopf



Arbeitsschritte		Zustand		q_H
Zustand	Symbol	Zustand	Schreibe	gehe nach
q_0	1	q_0	1	R
q_0	0	q_1	1	R
q_1	0	q_1	1	R
q_1	1	q_H	1	R

Kontrolleinheit

Berechenbarkeit

- Es gibt einen Algorithmus zur Berechnung
- Formal: Es gibt eine Turingmaschine zur Berechnung
- Welche Probleme sind mit Hilfe einer Maschine lösbar?
- Turing-Mächtigkeit bzw. Turing-Vollständigkeit
- Church-Turing-These: *Die Klasse der Turing berechenbaren Funktionen ist genau die Klasse der intuitiv berechenbaren Funktionen.*

Entscheidbarkeit

- Es gibt einen Algorithmus (Turingmaschine), der entscheidet, ob charakteristische Funktion der formalen Sprache berechenbar ist.
- Äquivalent: Kann ein Satz innerhalb der Theorie bewiesen werden oder nicht
- Halteproblem ist nicht entscheidbar
- vgl. Wortproblem