Theoretische Informatik

Alphabete, Wörter und Sprachen

Alphabete

Ein **Alphabet** Σ ist eine endliche, nichtleere Menge von Symbolen.

Sprachen

Eine **Sprache** L über einem Alphabet Σ ist eine Menge von Wörtern, die aus Symbolen von Σ bestehen. Eine Sprache kann endlich oder unendlich sein. Die leere Sprache wird mit $\emptyset$ bezeichnet.

Wörter

Ein **Wort** w ist eine endliche Folge von Symbolen aus einem Alphabet Σ. Die Länge eines Wortes w wird mit $|w|$ bezeichnet. Das leere Wort wird mit ε dargestellt und hat die Länge 0. Die Menge aller Wörter über einem Alphabet Σ wird mit Σ^* bezeichnet (Kleenesche Hülle).

Endliche Automaten

Deterministische endliche Automaten (DEA)

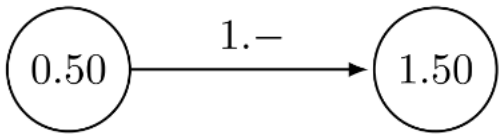
Ein DEA ist ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$, wobei:

- Q eine endliche Menge von Zuständen ist,
- Σ ein Alphabet ist,
- $\delta : Q \times \Sigma \to Q$ die Übergangsfunktion ist,
- $q_0 \in Q$ der Startzustand ist,
- $F \subseteq Q$ die Menge der Endzustände ist.

Übergangsfunktion: $\delta(q_0, a_1) = q_1$
Ein Wort $w \in \Sigma^*$ wird akzeptiert, wenn es von M verarbeitet wird und der Endzustand in F liegt.

Nichtdeterministische endliche Automaten (NEA)

Ein NEA ist ähnlich aufgebaut, aber die Übergangsfunktion δ kann mehrere Zustände für ein Symbol zurückgeben: $\delta : Q \times \Sigma \to 2^Q$. Ein NEA akzeptiert ein Wort, wenn es mindestens einen Pfad gibt, der das Wort vollständig verarbeitet und in einem Endzustand endet.



Kontextfreie Grammatiken

Definition

Eine **kontextfreie Grammatik** (KFG) ist ein 4-Tupel $G = (V, \Sigma, P, S)$, wobei:

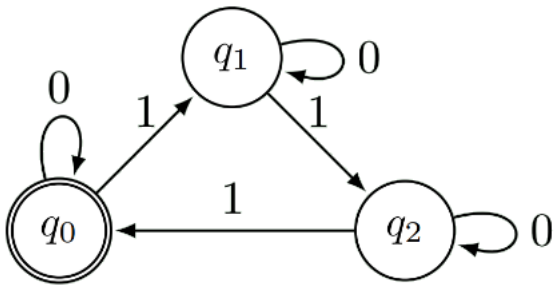
- V eine endliche Menge von Variablen ist,
- Σ ein Alphabet ist (Terminale),
- P eine endliche Menge von Produktionen ist,
- $S \in V$ das Startsymbol ist.

Produktionen

Eine Produktion hat die Form $A \to \alpha$, wobei $A \in V$ und $\alpha \in (V \cup \Sigma)^*$. Die Ableitung eines Wortes erfolgt durch wiederholtes Anwenden der Produktionen.

Ableitung

Ein Wort w wird aus G abgeleitet, wenn es durch eine endliche Folge von Produktionen aus dem Startsymbol S entsteht. Die Menge aller Wörter, die aus G abgeleitet werden können, bildet die Sprache $L(G)$ der Grammatik.



Kellerautomaten (KA)

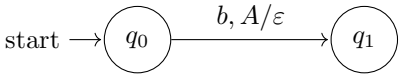
Ein **Kellerautomat** (Pushdown automata) (PDA) ist ein endlicher Automat, der zusätzlich einen Keller (Stack) hat. Er kann Symbole auf den Stack legen und entfernen. Ein Kellerautomat wird durch ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ definiert:

- Q eine endliche Menge von Zuständen,
- Σ ein Eingabealphabet,
- Γ ein Kelleralphabet,
- $\delta : Q \times \Sigma \times \Gamma \to 2^{Q \times \Gamma^*}$ die Übergangsfunktion,
- $q_0 \in Q$ der Startzustand,
- $Z_0 \in \Gamma$ das Startsymbol des Kellers,
- $F \subseteq Q$ die Menge der Endzustände.

Übergangsfunktion: $\delta(q_0, a_1, Z_0) = (q_1, Z_0 a_2)$

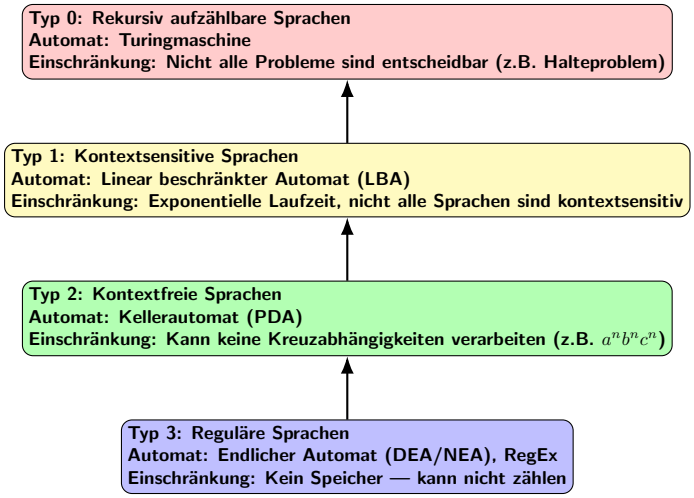
1. Ist im Zustand q_0
2. wird das Symbol a_1 gelesen,
3. wird das Symbol Z_0 vom Keller gepopt,
4. wechselt der Automat in den Zustand q_1,
5. und legt das Symbol $Z_0 a_2$ auf den Keller.

Ein Kellerautomat akzeptiert ein Wort, wenn es in einem Endzustand endet und der Keller leer ist.



Legende: $x, y/z$ bedeutet: lese x, poppe y vom Keller, pushe z auf den Keller.

Vergleich der Typen



Turingmaschinen

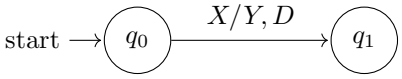
Eine **Turingmaschine** (TM) ist ein theoretisches Rechenmodell, das aus einem endlichen Steuerwerk, einem unendlichen Band und einem Lesekopf besteht. Sie kann Symbole auf dem Band lesen und schreiben sowie den Kopf bewegen.
Eine Turingmaschine wird durch ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ definiert:

- Q eine endliche Menge von Zuständen,
- Σ ein Eingabealphabet (ohne das leere Symbol),
- Γ ein Bandalphabet (enthält Σ und das leere Symbol),
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$ die Übergangsfunktion,
- $q_0 \in Q$ der Startzustand,
- $q_{accept} \in Q$ der Akzeptierzustand,
- $q_{reject} \in Q$ der Ablehnungszustand.

Übergangsfunktion: $\delta(q_0, X) = (q_1, Y, R)$ bedeutet:

1. Im Zustand q_0 wird das Symbol X gelesen,
2. der Zustand wechselt zu q_1,
3. das Symbol auf dem Band wird zu Y geändert,
4. der Lesekopf bewegt sich nach rechts (R).

Eine Turingmaschine akzeptiert ein Wort, wenn sie in den Zustand q_{accept} wechselt und das Band in einem akzeptierenden Zustand endet.



Legende: $x/y, D$ bedeutet: lese x, schreibe y auf das Band, bewege den Kopf in Richtung D (L für links, R für rechts, S für stehenbleiben).

Berechenbarkeit

Entscheidbare Probleme

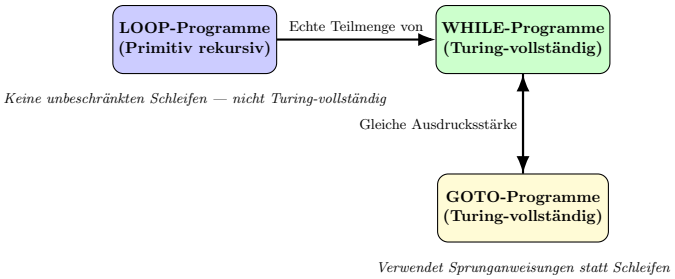
Ein Problem ist **entscheidbar**, wenn es eine Turingmaschine gibt, die für jede Eingabe in endlicher Zeit entscheidet, ob die Eingabe zur Sprache gehört oder nicht. Jede entscheidbare Sprache ist auch semi-entscheidbar.

semi-entscheidbare Probleme

Ein Problem ist **semi-entscheidbar**, wenn es eine Turingmaschine gibt, die für jede Eingabe, die zur Sprache gehört, in endlicher Zeit akzeptiert, aber für Eingaben, die nicht zur Sprache gehören, möglicherweise nicht terminiert.

Unentscheidbare Probleme

Ein Problem ist **unentscheidbar**, wenn es keine Turingmaschine gibt, die für alle Eingaben in endlicher Zeit entscheidet. Ein bekanntes Beispiel ist das Halteproblem.



Reduktion

Eine **Reduktion** ist eine Methode, um ein Problem A auf ein anderes Problem B zu transformieren. Wenn A auf B reduzierbar ist und B entscheidbar ist, dann ist auch A entscheidbar.
Wenn A auf B reduzierbar ist, schreiben wir $A \preceq B$. Reduktion ist transitiv.

Komplexitätstheorie

Komplexitätsklassen

- **P**: Klasse der Probleme, die in polynomieller Zeit gelöst werden können.
- **NP**: Klasse der Probleme, für die eine Lösung in polynomieller Zeit verifiziert werden kann.
- **NP-vollständig**: Die schwierigsten Probleme in NP, auf die alle anderen NP-Probleme in polynomieller Zeit reduziert werden können.
- **NP-schwer**: Probleme, die mindestens so schwer sind wie die NP-vollständigen Probleme, aber nicht notwendigerweise in NP liegen.

Polynomzeit-Verifizierer

Ein **Polynomzeit-Verifizierer** oder Zeuge ist ein Algorithmus, der eine Lösung für ein Problem in polynomieller Zeit überprüfen kann. Wenn ein Problem einen solchen Verifizierer hat, gehört es zur Klasse NP.

P und NP

Wenn es gelingt, ein NP-vollständiges Problem in polynomieller Zeit zu lösen, dann gilt $P = NP$. Aktuell ist unklar, ob $P = NP$ oder $P \neq NP$.

Landau-Symbole (Big-O-Notation)

- $\mathcal{O}(f(n))$: Obere Schranke für das Wachstum einer Funktion.
- $\Omega(f(n))$: Untere Schranke für das Wachstum einer Funktion.
- $\Theta(f(n))$: Exakte Schranke, wenn obere und untere Schranke gleich sind.
- $o(f(n))$: Strikte obere Schranke, wenn das Wachstum kleiner ist als $f(n)$.
- $\omega(f(n))$: Strikte untere Schranke, wenn das Wachstum größer ist als $f(n)$.