Verständnis formale Sprachen

Beantworten Sie kurz, präzise und mit Begründung folgende Fragen: (Die Begründungen müssen keine formellen mathematischen Beweise sein).

(a) Welche Möglichkeiten gibt es, eine formale Sprache vom Typ 3 zu definieren?

> - reguläre Grammatik
> - endlicher nichtdeterministische und deterministischer Automat
> - regulärer Ausdruck

(b) Was ist die Komplexität des Wortproblems für Typ-3 Sprachen und wieso ist das so?

> P, CYK-Algorithmus löst es ein Polynomialzeit.

(c) Sind Syntaxbäume zu einer Grammatik immer eindeutig? Falls nicht, geben Sie ein Gegenbeispiel.

> Nein. Syntaxbäume zu einer Grammtik sind nicht immer eindeutig.
>
> **Gegenbeispiel**
>
> $G = (\{S, A, B\}, \{a\}, P, S)$
>
> $P = \{$
>
> $$S \to AA$$
> $$S \to BB$$
> $$A \to a$$
> $$B \to a$$
>
> $\}$
>
> $S \vdash AA \vdash aA \vdash aa$
>
> $S \vdash BB \vdash aB \vdash aa$

(d) Wie kann man die Äquivalenz zweier Typ-3 Sprachen nachweisen?

> Myhill-Nerode-Äquivalenz. Wir können von den auf Äquivalenz zu überprüfenden Sprachen jeweils einen minimalen endlichen Automaten bilden. Sind diese entstanden zwei Automaten äquivalent so sind auch die Sprachen äquivalent.

(e) Wie kann man das Wortproblem für das Komplement einer Typ-3 Sprache lösen?

Da das Komplement einer regulären Sprache wieder eine reguläre Sprache ergibt, kann das Wortproblem beim Komplement durch einen deterministisch endlichen Automaten gelöst werden. Tausche akzeptierende mit nicht akzeptierenden Zuständen des zugehörigen Automaten.

Alternativ: Ergebnis des CYK-Algorithmus invertieren oder CYK auf Komplement, da reguläre Sprachen unter dem Komplement abgeschlossen sind.

(f) Weshalb gilt das Pumping-Lemma für Typ 3 Sprachen?

endliche Anzahl von Zuständen n im Automaten $\rightarrow$ für Wörter $|\omega| > n$ muss Zyklus vorhanden sein.

(g) Ist der Nachweis, dass das Typ-3 Pumping-Lemma für eine gegebene Sprache gilt, ausreichend, um zu zeigen, dass die Sprache vom Typ 3 ist? Falls nicht, geben Sie ein Gegenbeispiel, mit Begründung.

Nein:

Die Sprache $L = \{a^m b^n c^n \mid m, n \geq 1\} \cup \{b^m c^n \mid m, n \geq 0\}$ ist nicht regulär. Allerdings erfüllt L die Eigenschaften des Pumping-Lemmas, denn jedes Wort $z \in L$ lässt sich so zerlegen $z = uvw$, dass auch für alle $i \geq 0$ $uv^iw \in L$. Dazu kann v einfach als erster Buchstabe gewählt werden. Dieser ist entweder ein a, die Anzahl von führenden as ist beliebig. Oder er ist ein b oder c, ohne führende as ist aber die Anzahl von führenden bs oder cs beliebig. [a]

[a] https://de.wikipedia.org/wiki/Pumping-Lemma

(h) Geben Sie ein Beispiel, an dem deutlich wird, dass deterministische und nichtdeterministische Typ-2 Sprachen unterschiedlich sind.

Deterministisch Kontextfrei $L = \{\, 0^n 1^n \mid n \geq 0 \,\}$

Nichtdeterministisch Kontextfrei $L = \{\, \omega\omega^R \mid \omega \geq \{0,1\}^* \,\}$ (R steht für rückwärts)

[a]

[a] https://docplayer.org/19566652-Einfuehrung-in-die-theoretische-informatik.html

(i) Worin macht sich der Unterschied zwischen Typ 0 und 1 bemerkbar, wenn man Turingmaschinen benutzt, um das Wortproblem vom Typ 0 oder 1 zu lösen. Warum ist das so?

Typ 0: semi-entscheidbar, Typ 1: entscheidbar

Typ 0: Unendlichkeit des Band kann die unendlich lange Berechenbarkeit zustande kommen.

Typ 1: Linear beschränkte Turingmaschine endlich, dadurch Anzahl an Kombiniation

Typ 1 Sprachen sind monoton wachsend.

Da Typ 1 nur Wörter verlängert, kann daher in Polynomialzeit überprüft werden, ob das Wort in der Sprache liegt, indem die Regeln angewendet werden, bis das Wortende erreicht ist.