

Reguläre Sprachen, Ausdrucksstärke (Teil 2)

BC George (HSBI)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Wiederholung

Endliche Automaten, reguläre Ausdrücke, reguläre Grammatiken, reguläre Sprachen

- Wie sind DFAs und NFAs definiert?
- Was sind reguläre Ausdrücke?
- Was sind formale und reguläre Grammatiken?
- In welchem Zusammenhang stehen all diese Begriffe?

$\begin{matrix} \rightarrow & & \vdash & & \rightarrow \\ \cup & & \cup & & \cup \\ X \rightarrow aY \end{matrix}$

Motivation

Was haben reguläre Sprachen mit Compilern zu tun?

- ein Compiler muss
 - Schlüsselwörter
 - Operatoren
 - Bezeichner u. Zahlenverarbeiten.

Die werden mit regex beschrieben
und (oft) mit DFAs akzeptiert

- Reguläre Sprachen
- Lexer
- Grenzen regulärer Sprachen

Wozu reguläre Sprachen im Compilerbau?


Reguläre Ausdrücke

- definieren Schlüsselwörter und alle weiteren Symbole einer Programmiersprache, z. B. den Aufbau von Gleitkommazahlen
- werden (oft von einem Generator) in DFAs umgewandelt
- sind die Basis des *Scanners* oder *Lexers*

Lexer

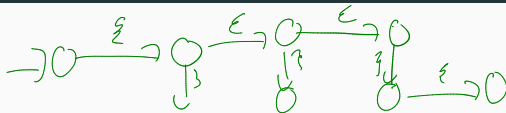
Ein Lexer ist mehr als ein DFA

Ein Lexer

- kann aus regulären Ausdrücken automatisch generiert werden
- wandelt mittels DFAs aus regulären Ausdrücken die Folge von Zeichen der Quelldatei in eine Folge von sog. Token um
- bekommt als Input eine Liste von Paaren aus regulären Ausdrücken und Tokennamen, z. B. (“while”, WHILE)
- Kommentare und Strings müssen richtig erkannt werden. (Schachtelungen)
- liefert Paare von Token und deren Werte, sofern benötigt, z. B. (WHILE, _), oder (IDENTIFIER, “radius”) oder (INTEGERZAHL, “334”)


Wofür reichen reguläre Sprachen nicht?

if ... {
if ε
if ε while ε }

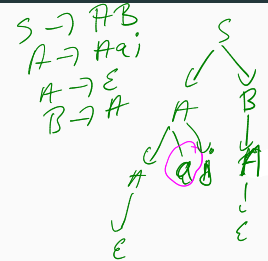


Für z. B. alle Sprachen, in deren Wörtern Zeichen über eine Konstante hinaus gezählt werden müssen. Diese Sprachen lassen sich oft mit Variablen im Exponenten beschreiben, die unendlich viele Werte annehmen können.

- $a^i b^{2*i}$ ist nicht regulär $a^* b^*$
- $a^i b^{2*i}$ für $0 \leq i \leq 3$ ist regulär $\varepsilon \mid abb \mid aabb \mid aaabbb$

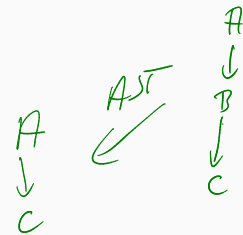
- Wo finden sich die oben genannten Variablen bei einem DFA wieder?
- Warum ist die erste Sprache oben nicht regulär, die zweite aber?

Wie geht es weiter?



Ein Parser

- führt mit Hilfe des Tokenstreams vom Lexer die Syntaxanalyse durch
- basiert auf einer sog. kontextfreien Grammatik, deren Terminale die Token sind
- liefert die syntaktische Struktur in Form eines Ableitungsbaums (**syntax tree**, **parse tree**), bzw. einen **AST** (abstract syntax tree) ohne redundante Informationen im Ableitungsbaum (z. B. Semikolons)
- liefert evtl. Fehlermeldungen
 - wo?
 - welcher?



Wrap-Up

- Definition und Aufgaben von Lexern
- Zusammenhänge zwischen diesen Mechanismen und Lexern, bzw. Lexergeneratoren
- Grenzen regulärer Sprachen



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Last modified: 9697eda (lecture: fix indentation/formatting (Regular), 2025-10-15)