Überführung regulärer Ausdrücke in endliche Automaten

Der Algorithmus von Thompson

Karin Haenelt

9.5.2010

Inhalt

- Quelle
- Prinzip des Algorithmus
- Algorithmus
- Konstruktion des Automaten
  - Basisausdrücke
  - Vereinigung, Konkatenation, Hülle
- Beispiel
- Implementierung
- Komplexität

Quelle

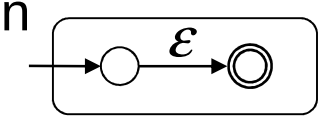
- Algorithmus entwickelt von Ken Thompson

- Ken Thompson (1968). Regular expression search algorithms. In: CACM 11(6): 419-422.

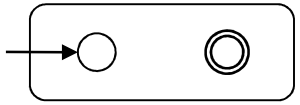
Prinzip des Algorithmus

- Angabe von Automaten für
  - elementare und
  - zusammengesetzte Ausdrücke
- Konstruktion durch Zusammensetzung aus Teilautomaten

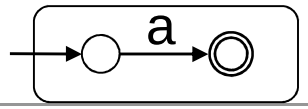
E = ε



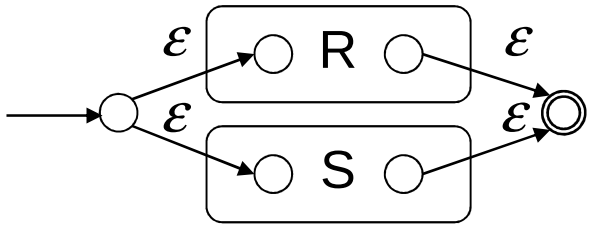
E = {}



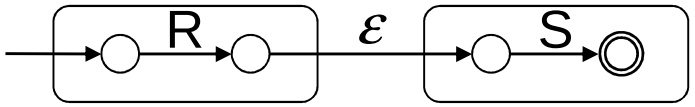
E = *a*



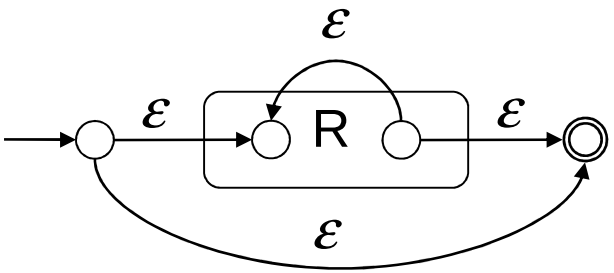
E=*R*|*S*



E=*RS*



E=*R**

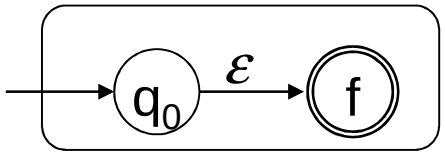


Algorithmus

Thompson	
Endesymbol bestimmen	
Erkennen des regulären Ausdrucks	Algorithmus von Hopcroft/Ullman 1988
Erzeugung eines ε-NEA	Erweiterung: Einfügen der Konstruktionsschritte für einen ε-NEA von Thompson 1968 in den Algorithmus von Hopcroft/Ullman 1988

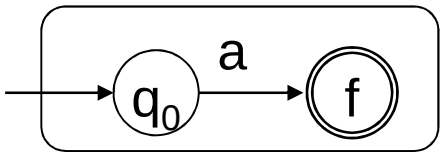
Konstruktion: Basisausdrücke

Basissymbol ε



E = ε

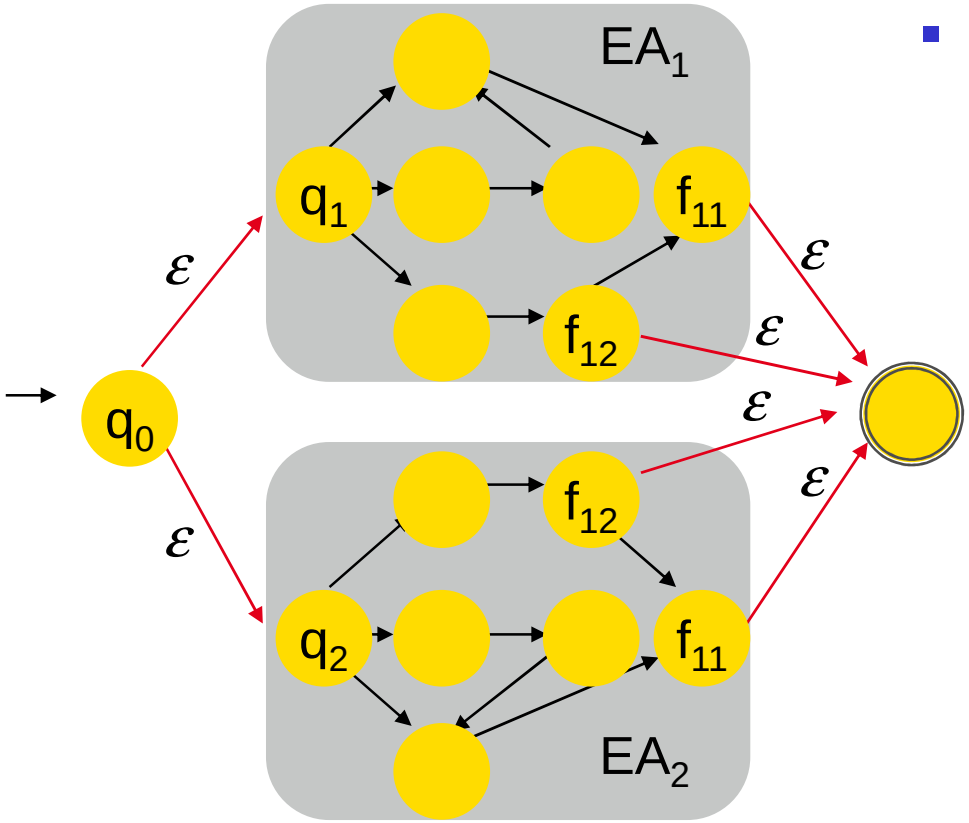
Basissymbol a



E = a

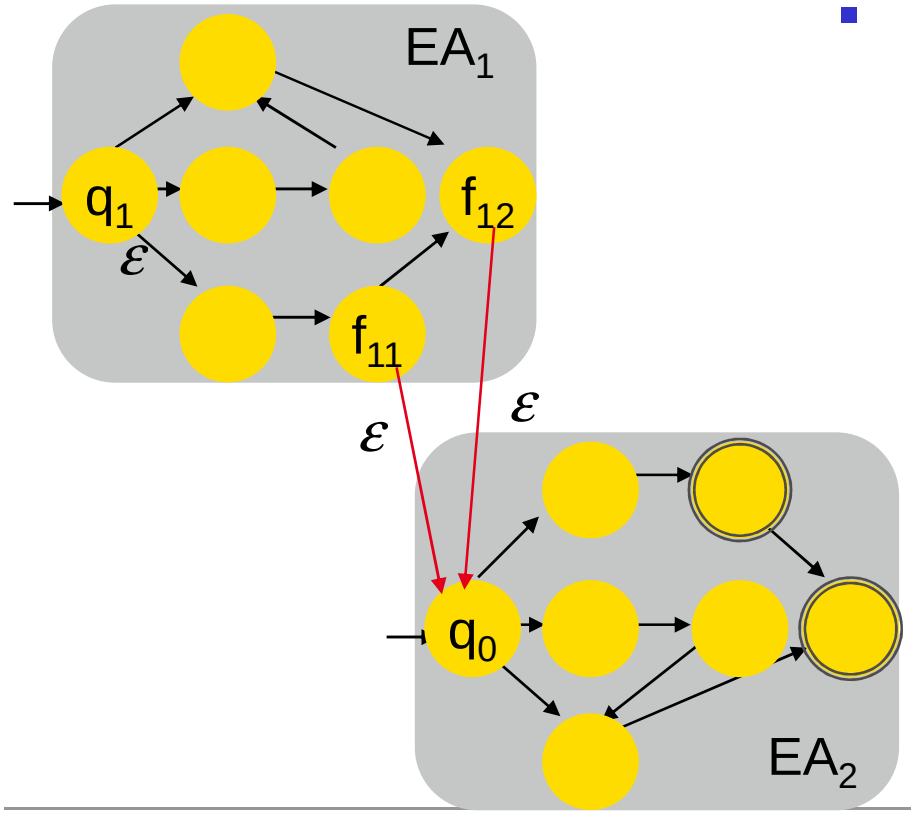
Basissymbol ε	erzeuge - einen Anfangszustand q_0 - einen Endzustand f - eine Kante mit dem Eingabesymbol ε von Zustand q_0 zu Zustand f
Basissymbol *a*	erzeuge - einen Anfangszustand q_0 - einen Endzustand f - eine Kante mit dem Eingabesymbol *a* von Zustand q_0 zu Zustand f

Konstruktion: Vereinigung



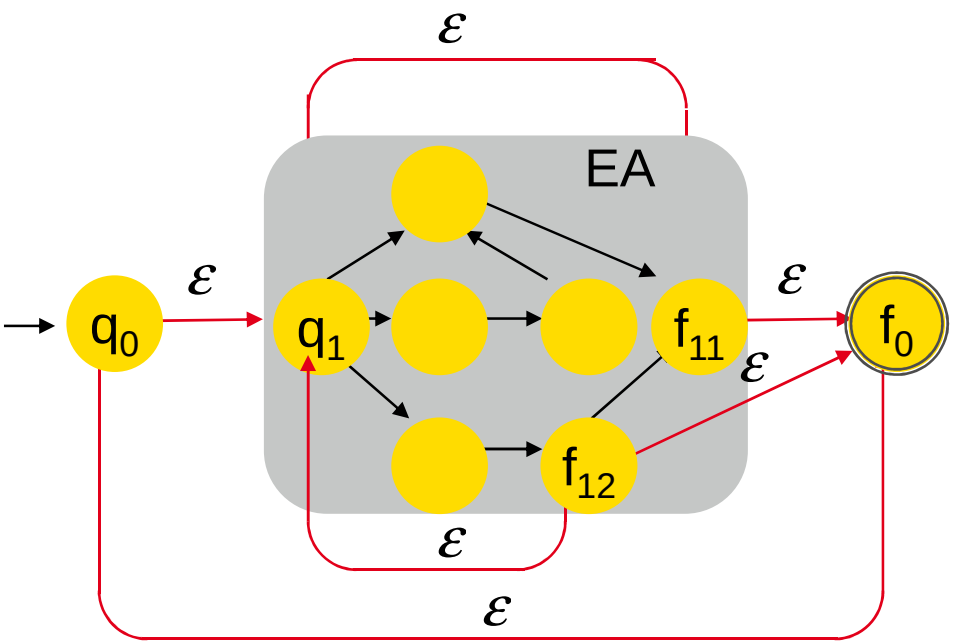
- erzeuge
  - Startzustand q_0
  - ε-Transition von q_0 zu den Startzuständen von EA_1 und EA_2
  - Endzustand f_0
  - ε-Transitionen von den Endzuständen von EA_1 und EA_2 zu f_0

Konstruktion: Konkatenation



- erzeuge
  - ε-Transitionen von *den* Endzuständen von EA_1 zum Startzustand von EA_2

Konstruktion: Hüllenbildung



- erzeuge
  - Startzustand q_0, Endzustand f_0
  - ε-Transition von q_0 nach q_1
  - ε-Transition von q_0 nach f_0 (modelliert Möglichkeit von null Vorkommen)
  - ε-Transitionen von $F = \{f_{11},f_{12},\ldots,f_{1n}\}$ nach q_1 (modelliert Wiederholung)
  - ε-Transition von $\{f_{11},f_{12},\ldots,f_{1n}\}$ nach f_0

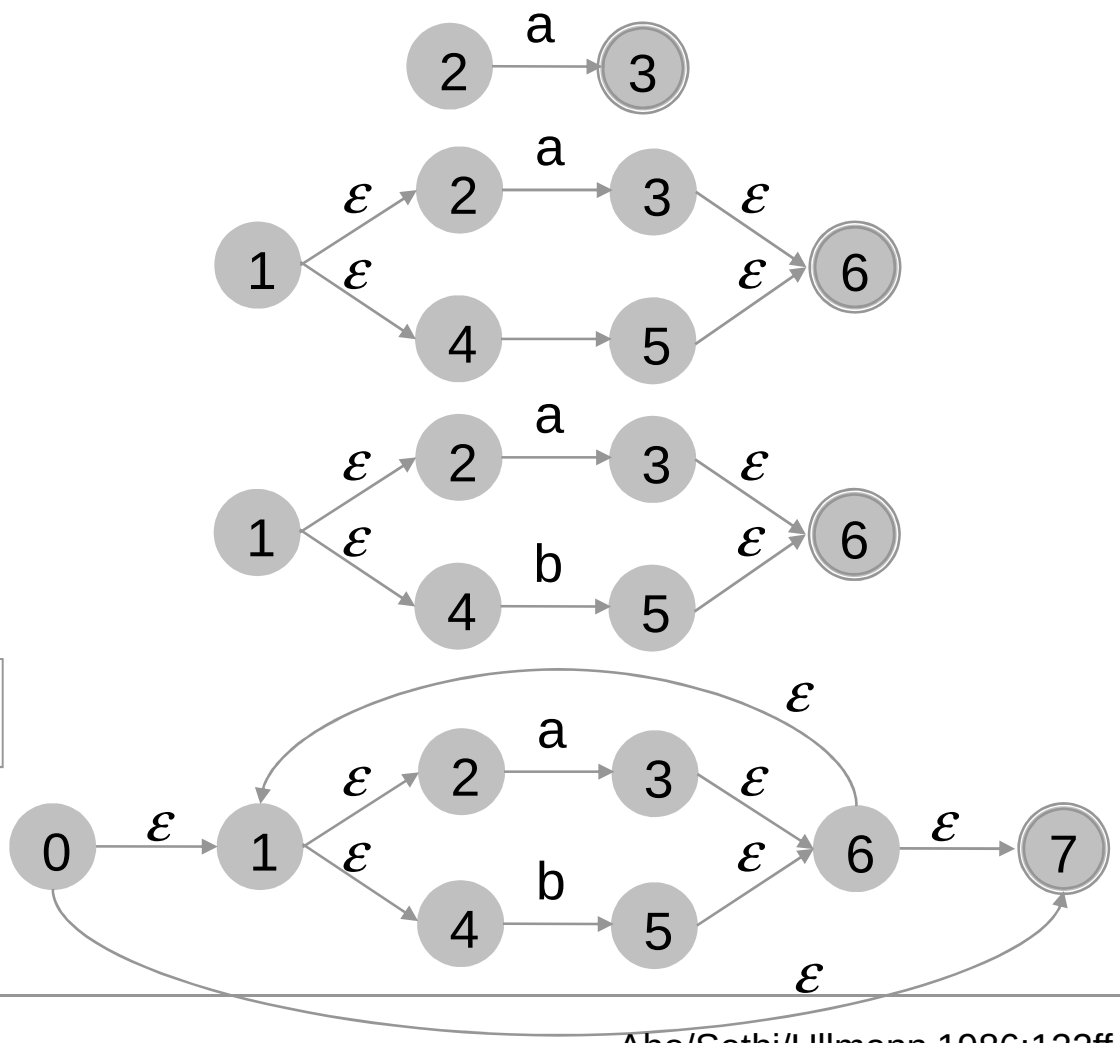
Beispiel

(a

(a|

(a|b)

(a|b)*



Aho/Sethi/Ullmann 1986:122ff

Algorithmus zur Erkennung regulärer Ausdrücke

nach Hopcroft/Ullmann 1988:128

```
1  procedure FINDE_AUSDRUCK;
     begin
3      FINDE_PRODUKT;
4      while  erste Symbol von STRING ist '+' do // Disjunktion
7              { stringIndex++; FINDE_PRODUKT; }
     end FINDE_AUSDRUCK;

   procedure FINDE_PRODUKT;
     begin
12     FINDE_TERM;
       while   erstes Symbol von String ist '•' do // Konkatenation
16             { stringIndex++; FINDE_TERM; }
     end FINDE_PRODUKT;

   procedure FINDE_TERM;
     begin
21     if          erstes Symbol von STRING ist TERMINAL then { stringIndex++;}
23     else if     erstes Symbol von STRING ist '(' then
26                 {       stringIndex++; FINDE_AUSDRUCK;
27                         if          erstes Symbol von STRING ist ')' then stringIndex++;
                           else        Fehler }
31     while       erstes Symbol von STRING ist '*' do // Kleenesche Hülle
                   {stringIndex++;}
     end FINDE_TERM;
```

Konstruktion des Automaten -1-

Hopcroft/Ullman: Erweiterung des Erkenners um Konstruktionsanweisungen:

- Prozedur FINDE_TERM:
  - Basisausdruck: Erzeugung eines Automaten, der den Basisausdruck akzeptiert
  - (: Automat, der durch FINDE_AUDRUCK zurückgegeben wird, ist der Wert von FINDE_TERM
  - *: Modifikation des Automaten, so dass er die Kleenesche Hülle akzeptiert

Hopcroft/Ullmann 1986:128/129

Konstruktion des Automaten -2-

- Prozedur FINDE_PRODUKT:
  - FINDE_PRODUKT wird der Wert des ersten Aufrufs von FINDE_TERM zugewiesen
  - Konkatenation: Bei Ausführung der while-Anweisung wird der Wert von FINDE_PRODUKT auf einen Automaten gesetzt, der die Konkatenation der Mengen akzeptiert, die durch den aktuellen Wert von FINDE_PRODUKT und dem Automaten akzeptiert werden, der durch den Aufruf von FINDE_TERM in der while-Schleife zurückgegeben wird.
- Prozedur FINDE_AUSDRUCK (Erweiterung um ähnliche Anweisungen)

Hopcroft/Ullmann 1986:128/129

Komplexität

- Größe der Automaten wächst linear mit der Länge der regulären Ausdrücke
- Konstruktion des ε-NEA in linearer Zeit
- Eliminierung der ε-Transitionen in quadratischer Zeit (Aho/Sethi/Ullman, 1986) (Hopcroft/Ullman, 1988)

Literatur

- Quellen
  - Thompson, Ken (1968). Regular expression search algorithms. In: CACM 11(6): 419-422.
- Erläuterungen
  - Hopcroft, John E. und Jeffrey D. Ullman. *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie.* Bonn u. a. : Addison-Wesley, 1988 (engl. Original Introduction to automata theory, languages and computation).
  - Hopcroft, John E., Rajeev Motwani und Jeffrey D. Ullman (2002). *Einführung in die Automatentheorie, Formale Sprachen und Komplexität*. Pearson Studium
    engl. Original: *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley. www-db.stanford.edu/~ullman/ialc.html
- weitere Literatur
  - Aho, Alfred V.; Sethi, Ravi und Jeffrey D. Ullman (1986). *Compilers. Principles, Techniques and Tools*. Addison-Wesley Publishing Company.

Versionen

- 09.05.2010
- 05.05.2007
- 27.05.,20.03.2005