RWTH Aachen
Lehrstuhl für Informatik VII
Prof. Dr. Dr.h.c. Wolfgang Thomas

Wintersemester 2005/06

Seminar über „Automaten für XML“

DTDs mit einfachen regulären Ausdrücken

Mark Wiesemann

1. Februar 2006
Betreuer: Philipp Rohde

Inhaltsverzeichnis

1 Einleitung

XML-Dokumente sind über Dokumenttypdefinitionen (DTDs) definiert. DTDs können als erweiterte kontextfreie Grammatiken interpretiert werden, auf deren rechten Regelseiten reguläre Ausdrücke vorkommen können. In der Praxis kommen in den meisten Fällen aber nur *einfache* reguläre Ausdrücke (vgl. Definition 2) in Dokumenttypdefinitionen vor.

Beispiel 1.
`library` $\rightarrow$ `book`*
`book` $\rightarrow$ `title author`$^+$ `date isbn`
`date` $\rightarrow$ `month year`

□

Die Analyse von XML-Dokumenten und DTDs erfordert das Lösen von Entscheidungsproblemen wie dem Inklusions-, dem Äquivalenz- und dem Schnittproblem von regulären Ausdrücken. Im Allgemeinen sind diese Entscheidungsprobleme nicht effizient lösbar, da sie PSPACE-vollständig sind (vgl. Abschnitt 1.3).

1.1 Reguläre Ausdrücke

Definition 1 (reguläre Ausdrücke über Σ).
Sei Σ ein endliches Alphabet. REGULÄRE AUSDRÜCKE über Σ sind spezielle Formeln, mit denen Sprachen definiert werden können. Sie sind wie folgt induktiv definiert:

- $\varnothing$, ϵ, a für $a \in \Sigma$ sind reguläre Ausdrücke.
- Wenn r und s reguläre Ausdrücke sind, dann sind auch rs, $r + s$ und r^* reguläre Ausdrücke.

Wichtige Abkürzungen für reguläre Ausdrücke sind:

- $r?$ für $r + \epsilon$
- r^+ für rr^*

Die von regulären Ausdrücken definierten Sprachen sind:

- $\mathcal{L}(\varnothing) = \varnothing$, $\mathcal{L}(\epsilon) = \{\epsilon\}$, $\mathcal{L}(a) = \{a\}$ für $a \in \Sigma$
- $\mathcal{L}(rs) = \{vw \mid v \in \mathcal{L}(r), w \in \mathcal{L}(s)\}$
- $\mathcal{L}(r + s) = \mathcal{L}(r) \cup \mathcal{L}(s)$
- $\mathcal{L}(r^*) = \{\epsilon\} \cup \bigcup_{i=1}^{\infty} \mathcal{L}(r)^i$ □

Definition 2 (Klassen von einfachen regulären Ausdrücken).
Sei Σ ein endliches Alphabet. Dann ist die KLASSE DER EINFACHEN REGULÄREN AUSDRÜCKE mit den Fragmenten a und a^* (im Folgenden abgekürzt durch $\mathrm{RE}(a, a^*)$) gleich $\{e_1 \dots e_n \mid e_i = a \text{ oder } e_i = a^* \text{ für ein } a \in \Sigma\}$. Die KLASSE DER EINFACHEN REGULÄREN AUSDRÜCKE mit den Fragmenten a und a^+ (im Folgenden abgekürzt durch $\mathrm{RE}(a, a^+)$ ist gleich $\{e_1 \dots e_n \mid e_i = a \text{ oder } e_i = a^+ \text{ für ein } a \in \Sigma\}$. Jedes e_i wird als FAKTOR bezeichnet. □

Beispiel 2.
(1) In $\mathrm{RE}(a, a^*)$ liegen Ausdrücke wie a^*, a^*b und ab^*c^*d.

(2) In $\mathrm{RE}(a, a^+)$ liegen Ausdrücke wie ab^+cd, ab und b^+c. □

1.2 Dokumenttypdefinition

Definition 3 (DTD).
Eine DOKUMENTTYPDEFINITION (DTD, engl.: *document type definition*) ist ein Paar (d, s_d). d ist eine Funktion, die Σ-Symbole auf reguläre Ausdrücke über Σ abbildet, $s_d \in \Sigma$ ist das Startsymbol. (d, s_d) wird häufig durch d abgekürzt. Sie kann als erweiterte kontextfreie Grammatik gesehen werden.

Die Sprache $\mathcal{L}(d)$ ist induktiv definiert. In ihr sind die Bäume, die nur aus der Wurzel bestehen und mit a beschriftet sind, enthalten, falls $\varepsilon \in d(a)$. Seien das Wort $a_1 \dots a_n \in d(a)$ und Bäume $t_i \in \mathcal{L}(d)$ für $1 \leq i \leq n$ gegeben, wobei a_i die Beschriftung der Wurzel von t_i ist. Dann ist ebenfalls der Baum $a(t_1, \dots, t_n) \in \mathcal{L}(d)$. □

Eine DTD beschreibt die Struktur eines XML-Dokuments. Der Grundaufbau von DTDs ist (vgl. [HMU02]):

```
<!DOCTYPE Name-der-DTD [
  Liste der Elementdefinitionen
]>
```

In DTDs werden Elemente, die auch als Tags bezeichnet werden, wie folgt definiert:

```
<!ELEMENT Elementname (Beschreibung des Elements)>
```

Bei den Beschreibungen der Elemente handelt es sich um eine Art von regulären Ausdrücken. Sie können sowohl aus Namen anderer Elemente als auch aus dem Wert `#PCDATA` bestehen, der für beliebigen Text steht und keine weiteren XML-Elemente enthalten darf.

Elemente können durch `|` vereinigt werden. So bedeutet
`<!ELEMENT disk (harddisk | cd | dvd)>`,
dass ein `disk`-Element als Unterelement entweder ein `harddisk`-, ein `cd`- oder ein `dvd`-Element enthält, nicht aber gleichzeitig mehrere dieser Unterelemente. In Grammatik-Schreibweise entspricht diese DTD-Zeile der Regel `disk` → `harddisk|cd|dvd`.

Soll ein Element mehrere Unterelemente enthalten, so müssen die Elementnamen durch Kommata getrennt werden. Die Elementdefinition
`<!ELEMENT processor (manufacturer, model, speed)>` bedeutet also, dass ein `processor`-Element als Unterelemente sowohl ein `manufacturer`-, ein `model`- als auch ein `speed`-Element enthalten muss. Dieses Element entspricht der Grammatik-Schreibweise `processor` → `manufacturer model speed`.

Hinter den Namen der Elemente können drei verschiedene Operatoren zur Hüllenbildung gesetzt werden. Diese sind: `*`, um auszudrücken, dass ein Element beliebig oft vorkommt. `+`, um auszudrücken, dass ein Element mindestens einmal vorkommt. `?`, um auszudrücken, dass ein Element entweder keinmal oder einmal vorkommt.

Beispiel 3.
Der in der Einführung gezeigten Beispiel-DTD in Grammatik-Schreibweise (vgl. Beispiel 1) entspricht folgende DTD-Schreibweise:

```
<!DOCTYPE library [
  <!ELEMENT library (book*)>
  <!ELEMENT book (title, author+, date, isbn)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT date (month, year)>
  <!ELEMENT isbn (#PCDATA)>
  <!ELEMENT month (#PCDATA)>
  <!ELEMENT year (#PCDATA)>
]>
```

□

Beispiel 4.
Zu der in Beispiel 3 gezeigten DTD ist folgendes XML-Dokument gültig:

```
<library>
  <book>
    <title>XML in a Nutshell</title>
    <author>Harold</author>
    <author>Means</author>
    <date>
      <month>1</month>
      <year>2005</year>
    </date>
    <isbn>3-89721-339-7</isbn>
  </book>
  <book>
    ...
  </book>
</library>
```

□

1.3 Entscheidungsprobleme

Definition 4 (Entscheidungsprobleme).
Sei $\mathcal{R}$ eine Klasse von regulären Ausdrücken.

Das INKLUSIONSPROBLEM für $\mathcal{R}$ ist, für $r, r' \in \mathcal{R}$ zu entscheiden, ob $\mathcal{L}(r) \subseteq \mathcal{L}(r')$ gilt.

Das ÄQUIVALENZPROBLEM für $\mathcal{R}$ ist, für $r, r' \in \mathcal{R}$ zu entscheiden, ob $\mathcal{L}(r) = \mathcal{L}(r')$ gilt.

Das SCHNITTPROBLEM für $\mathcal{R}$ ist, für $r_1, \ldots, r_n \in \mathcal{R}$ (mit $n \in \mathbb{N}$) zu entscheiden, ob $\bigcap_{i=1}^{n} \mathcal{L}(r_i) \neq \varnothing$ gilt. □

Bemerkung 1.
Diese drei Entscheidungsprobleme sind für allgemeine reguläre Ausdrücke PSPACE-vollständig. □

	Inklusion	Äquivalenz	Schnitt
a, a^+	PTIME	PTIME	PTIME
a, a^*	CONP-vollst.	PTIME	NP-vollst.
$a, a?$	CONP-vollst.	PTIME	NP-vollst.
$a, (+a)^*$	PSPACE-vollst.	PSPACE	NP-vollst.
$\text{RE}^{\leq k}$ $(k \geq 3)$	PTIME	PTIME	PSPACE-vollst.

Tabelle 1: Von Martens, Neven und Schwentick untersuchte Komplexitäten regulärer Ausdrücke

Martens, Neven und Schwentick haben in [MNS04] die in Tabelle 1 dargestellten Komplexitäten für Fragmente einfacher regulärer Ausdrücke bestimmt. Dabei ist $(+a)^*$ die Abkürzung für $(a_1 + a_2 + \ldots + a_n)^*$ und $\text{RE}^{\leq k}$ bedeutet, dass jedes Symbol aus Σ nur maximal k-mal in einem einfachen regulären Ausdruck vorkommt.

In den beiden folgenden Abschnitten soll gezeigt werden, warum das Äquivalenzproblem für $\text{RE}(a, a^*)$ in PTIME liegt (vgl. Abschnitt 2) und warum das Schnittproblem für $\text{RE}(a, a^*)$ NP-vollständig ist (vgl. Abschnitt 3).

2 Äquivalenzproblem für einfache reguläre Ausdrücke

Theorem 1.
Für $\text{RE}(a, a^*)$ liegt das Äquivalenzproblem in PTIME. □

Die Idee für den Beweis ist, eine Normalform für reguläre Ausdrücke aus dieser Klasse zu definieren, die in PTIME berechnet werden kann. Diese Normalform sei als starke Sequenz-Normalform bezeichnet. Im Beweis wird gezeigt, dass zwei reguläre Ausdrücke aus $\text{RE}(a, a^*)$ genau dann äquivalent sind, wenn sie die gleiche starke Sequenz-Normalform haben.

Definition 5 (Sequenz-Normalform).
Sei r ein regulärer Ausdruck aus $\text{RE}(a, a^*)$ und sei $d \in \Sigma$. Dann gibt $d[i, j]$ an, dass d mindestens i-mal und höchstens j-mal direkt nacheinander vorkommt. $j = *$ gibt dabei an, dass die Anzahl der Wiederholungen von d unbeschränkt ist.

$d[1, 1]$ steht für d, $d[0, 1]$ steht für $d?$ und $d[0, *]$ steht für d^*.

Durch Zusammenfassung von direkt aufeinander folgenden $d[i_1, j_1]$ und $d[i_2, j_2]$ zu $d[i_1 + i_2, j_1 + j_2]$, wann immer möglich, entsteht die SEQUENZ-NORMALFORM. Dabei ist $j + * = * + j = *$ für $j \in \mathbb{N} \cup \{*\}$. □

Beispiel 5 (Sequenz-Normalform).
Die Sequenz-Normalform des regulären Ausdrucks

$$aa^*aa^*b^*bb^*b^*a$$

ist $a[2, *]b[1, *]a[1, 1]$. □

In einigen Fällen ist diese Normalform nicht ausreichend, um Äquivalenz zu bestimmen, da z.B. die regulären Ausdrücke

$$r_1(a,b,i,l) = a[i,*]b[0,*]a[0,*]b[1,*]a[l,*]$$
$$r_2(a,b,i,l) = a[i,*]b[1,*]a[0,*]b[0,*]a[l,*]$$

äquivalent sind, aber verschiedene Sequenz-Normalformen haben.

Um diese Äquivalenz zu zeigen, zeige: $w \in r_1 \Rightarrow w \in r_2$. Sei $w = a^{k_1}b^{k_2}a^{k_3}b^{k_4}a^{k_5}$.

(1) Seien $k_2 \geq 1, k_3 \geq 1$. Dann: $w \in r_2$.

(2) Seien $k_2 \geq 1, k_3 = 0$. Dann: $w = a^{k_1}b^{k_2+k_4}a^{k_5} \in r_2$.

(3) Seien $k_2 = 0, k_3 \geq 1$. Dann: $w = a^{k_1+k_3}b^{k_4}a^{k_5} \in r_2$.

(4) Seien $k_2 = 0, k_3 = 0$. Dann: $w = a^{k_1}b^{k_4}a^{k_5} \in r_2$.

Die Umkehrung ($w \in r_2 \Rightarrow w \in r_1$) folgt analog.

Definition 6 (starke Sequenz-Normalform).
Sei r ein regulärer Ausdruck aus $\mathrm{RE}(a, a^*)$. Dann entsteht r' durch Ersetzen von $r_1(a,b,i,l)$ durch $r_2(a,b,i,l)$ (mit $a, b \in \Sigma, i, l \in \mathbb{N}$; vgl. Beispiel 5). Wird diese Ersetzung so oft wie möglich durchgeführt, ist r' in STARKER SEQUENZ-NORMALFORM. □

r' ist unabhängig von der Reihenfolge der Ersetzungen und eindeutig. Die starke Sequenz-Normalform zu einem Ausdruck r sei im Folgenden mit $\mathrm{sSNF}(r)$ bezeichnet.

Definition und Notation 7.
Im folgenden Beweis werden die folgenden Notationen und Begriffe benötigt:

- Wenn f ein Ausdruck $\tau[i,j]$ ist, schreiben wir $e(f)$ für τ, $u(f)$ für die obere Grenze j und $l(f)$ für die untere Grenze i.
- Sei $r = r_1 \ldots r_n$ ein regulärer Ausdruck in Sequenz-Normalform.
  Dann ist $\max(r)$ $\begin{cases} \text{undefiniert} & \text{, falls alle } u(r_i) = * \\ = \max\{u(r_i) \mid u(r_i) \neq *\} & \text{, sonst} \end{cases}$.
- Sei a ein beliebiges Zeichen und w eine Zeichenkette. Dann heißt eine maximale Teil-Zeichenkette v von w, die die Form a^k hat, ein BLOCK von w. □

BEWEIS (ZU THEOREM 1).
Zu zeigen ist, dass zwei reguläre Ausdrücke aus $\mathrm{RE}(a, a^*)$ genau dann äquivalent sind, wenn sie die gleiche starke Sequenz-Normalform haben.

Beweise zunächst die Rückrichtung. Dazu seien r und s zwei reguläre Ausdrücke aus $\mathrm{RE}(a, a^*)$ mit gleicher starker Sequenz-Normalform. Da nach Konstruktion $\mathcal{L}(r) = \mathcal{L}(\mathrm{sSNF}(r))$ für jedes r in $\mathrm{RE}(a, a^*)$ gilt, folgt:

$$\mathcal{L}(r) = \mathcal{L}(\mathrm{sSNF}(r)) = \mathcal{L}(\mathrm{sSNF}(s)) = \mathcal{L}(s)$$

Daher sind r und s äquivalent.

Für die Hinrichtung sollen die folgenden Voraussetzungen gelten:

- Seien $r = r_1 \dots r_n$ und $s = s_1 \dots s_m$ aus $\mathrm{RE}(a, a^*)$ zwei äquivalente reguläre Ausdrücke in starker Sequenz-Normalform.
- Seien zunächst sowohl $\max(r)$ als auch $\max(s)$ definiert und sei k um eins größer als jede obere Grenze, die kein $*$ ist, in r und s, d.h. $k = 1 + \max\{\max(r), \max(s)\}$.

Beweise nun, dass $n = m$ und der i-te Faktor von r und der i-te Faktor s aus dem gleichen Symbol aus Σ bestehen ($e(r_i) = e(s_i)$), dass die oberen Grenzen dieser Faktoren gleich sein müssen ($u(r_i) = u(s_i)$) und dass die unteren Grenzen dieser Faktoren gleich sein müssen ($l(r_i) = l(s_i)$). Dann folgt $\mathrm{sSNF}(r) = \mathrm{sSNF}(s)$.

Zunächst wird die Gleichheit der Symbole ($e(r_i) = e(s_i)$) und $n = m$ bewiesen. Bilde dazu Wörter $v = v_1 \dots v_n$ und $w = w_1 \dots w_m$. Die Blöcke v_i bzw. w_i seien dabei wie folgt definiert:

$$v_i = \begin{cases} e(r_i)^k & , u(r_i) = * \\ e(r_i)^{u(r_i)} & , \text{sonst} \end{cases} \qquad w_i = \begin{cases} e(s_i)^k & , u(s_i) = * \\ e(s_i)^{u(s_i)} & , \text{sonst} \end{cases}$$

Da v nach dieser Konstruktion in $\mathcal{L}(r)$ liegt, folgt wegen der Äquivalenz-Annahme von r und s, dass auch $v \in \mathcal{L}(s)$ gilt. Da weiterhin v aus n Blöcken besteht (sonst wäre v nicht in Normalform), folgt, dass s mindestens n Faktoren haben muss. Analog kann für w argumentiert werden. Daher gilt $m = n$ und somit auch $e(r_i) = e(s_i)$.

Beweise nun, dass auch die oberen Grenzen der Faktoren gleich sein müssen ($u(r_i) = u(s_i)$). Wenn $u(r_i) = *$ gilt, dann muss v_i nach Konstruktion von einem Faktor mit oberer Grenze $*$ in s gematcht werden. Also: $u(s_i) = *$. Das analoge Argument gilt für $u(s_i) = *$, also folgt $u(r_i) = *$. Zusammen ergibt sich: $u(r_i) = * \Leftrightarrow u(s_i) = *$. Analog kann für Grenzen $< *$ argumentiert werden, so dass insgesamt folgt: $u(r_i) = u(s_i)$. Die Fälle, dass $\max(r)$ oder $\max(s)$ oder beide nicht definiert sind, werden analog behandelt. Zeige dazu ähnlich wie oben, dass $u(r_i) = * \Leftrightarrow u(s_i) = *$ gilt.

Abschließend muss gezeigt werden, dass auch die unteren Grenzen der Faktoren gleich sind. Bilde dazu die Wörter $v' = v'_1 \dots v'_n$ und $w' = w'_1 \dots w'_m$ mit $v'_i = e(r_i)^{l(r_i)}$ und $w'_i = e(s_i)^{l(s_i)}$. Analog zum Beweis der Gleichheit der oberen Grenzen folgt, dass $l(r_i) = l(s_i)$ für untere Grenzen > 0 gilt.

Für untere Grenzen $= 0$ wird nun ein Widerspruchsbeweis geführt. Dazu sei $l(r_i) < l(s_i)$ und i minimal mit dieser Bedingung. Sei zunächst $l(r_i) > 0$ und das Wort v'' sei wie v definiert, wobei der Block v_i durch $e(r_i)^{l(r_i)}$ ersetzt wird. Dann kann s nicht v'' matchen, weil die untere Grenze von s_i größer als die untere Grenze von r_i ist.

Sei nun $0 = l(r_i)$ (und weiterhin $l(r_i) < l(s_i)$). Nehme zunächst an, dass $l(s_i) \geq 2$ und dass das Wort w'' wie w definiert ist, wobei w_i durch $e(r_i)$ ersetzt wird. Dann matcht s w'' nicht, weil die untere Grenze von s_i größer als 1 ist ($l(s_i) \geq 2$), was ebenfalls zu dem gewünschten Widerspruch führt.

Nun sei $l(s_i) = 1$ (also: $0 = l(r_i) < l(s_i) = 1$). i muss zwischen 1 und n liegen, da s mit $i = 1$ nicht das Wort $v_2 \dots v_n$ ($v_1 = \epsilon$, da $l(r_1) = 0$) matcht, und da s mit $i = n$ nicht das Wort $v_1 \dots v_{n-1}$ ($v_n = \epsilon$, da $l(r_n) = 0$) matcht. Betrachte nun die beiden Fälle $e(r_{i-1}) \neq e(r_{i+1})$ und $e(r_{i-1}) = e(r_{i+1})$. Im ersten Fall wird ein Wort mit $n - 1$ Blöcken betrachtet. Hier ergibt sich der gewünschte Widerspruch, weil $e(s_i) \neq e(s_{i-1})$ und $e(s_i) \neq e(s_{i+1})$

und somit s_i nur ein v_j mit $j > i+1$ oder $j < i-1$ matchen könnte. Dann müssten aber mindestens i Blöcke vor bzw. hinter v_j von weniger Faktoren aus s (d.h. $s_1 \dots s_{i-1}$ bzw. $s_{i+1} \dots s_n$) gematcht werden. Im zweiten Fall ist r nicht mehr in starker Sequenz-Normalform, da r einen Ausdruck der Form $a[i,*]b[0,*]a[0,*]b[1,*]a[l,*]$ ($a, b \in \Sigma, i, l \in \mathbb{N}$) enthält. Somit ergibt sich auch hier der gewünschte Widerspruch.

Weil die Umformung eines einfachen regulären Ausdrucks in die starke Sequenz-Normalform in PTIME möglich ist und weil die Prüfung auf Gleichheit von zwei einfachen regulären Ausdrücken in starker Sequenz-Normalform ebenfalls in PTIME möglich ist, liegt das Äquivalenzproblem für $\mathrm{RE}(a, a^*)$ in PTIME. ■

3 Schnittproblem für einfache reguläre Ausdrücke

Theorem 2.
Für $\mathrm{RE}(a, a^*)$ ist das Schnittproblem NP-vollständig. □

Um die NP-Härte dieses Problems zu zeigen, wird eine Reduktion vom Erfüllbarkeitsproblem 3-CNF durchgeführt. Anschließend wird ein NP-Algorithmus gezeigt, um die NP-Vollständigkeit zu beweisen.

Definition 8 (Erfüllbarkeitsproblem / 3-CNF).
Das ERFÜLLBARKEITSPROBLEM 3-CNF ist:

Gegeben: Eine aussagenlogische Formel Φ in konjunktiver Normalform mit genau drei Literalen pro Klausel.

Frage: Ist Φ erfüllbar, d.h. gibt es eine Belegung der Variablen mit Werten aus $\{0, 1\}$, so dass Φ den Wert 1 erhält? □

Bemerkung 2.
3-CNF ist NP-vollständig. (Beweis: Satz von Cook, Reduktion von SAT auf 3-CNF) □

Theorem 3.
Für $\mathrm{RE}(a, a^+)$ liegt das Schnittproblem in PTIME. □

BEWEIS.
Seien $r_1, \dots, r_n$ aus $\mathrm{RE}(a, a^+)$ in Sequenz-Normalform.

Der Schnitt $r_1 \cap \dots \cap r_n$ kann nur $\neq \varnothing$ sein, wenn alle r_i die gleiche Anzahl m an Faktoren haben und wenn, für alle $j \leq m$, der j-te Faktor jedes r_i dasselbe Basissymbol a_j hat. Also gilt: $r_i = e_1^i \dots e_n^i$ mit $e_j^i = a_j[k_j^i, l_j^i]$ für $k_j^i, l_j^i \in \mathbb{N}$ und $k_j^i \geq 1$.

Seien nun $p_j := \max\{k_j^i \mid i \leq n\}$ und $q_j := \min\{l_j^i \mid i \leq n\}$ für alle $j \leq m$. Dann gilt: $r_1 \cap \dots \cap r_n \neq \varnothing \Leftrightarrow p_j \leq q_j$ für alle $j \leq m$. Der Schnitt ist also genau dann nicht leer, wenn für jedes $j \leq m$ die größte untere Grenze kleiner (oder gleich) als die kleinste obere Grenze ist.

Somit ist in PTIME entscheidbar, ob der Schnitt $r_1 \cap \dots \cap r_n$ nicht leer ist. ■

BEWEIS (ZU THEOREM 2).
Für die NP-Härte des Schnittproblems muss gezeigt werden, dass eine polynomielle Reduktion von 3-CNF existiert. Dazu sei $\Phi = C_1 \wedge \ldots \wedge C_k$ eine 3-CNF-Formel mit Variablen aus $\{x_1, \ldots, x_n\}$. Jedes der C_i ($1 \leq i \leq k$) sei definiert als $C_i = L_{i,1} \vee L_{i,2} \vee L_{i,3}$ mit $L_{i,j} = x_l$ oder $L_{i,j} = \neg x_l$ für ein l mit $1 \leq l \leq n$.

Wir konstruieren im Folgenden reguläre Ausdrücke $R_1, \ldots, R_k, S_1, S_2$, so dass gilt:

- R_i kodiert die Klausel C_i.
- S_1, S_2 kodieren zusammen Belegungen der Variablen von Φ, so dass $w \in \mathcal{L}(S_1) \cap \mathcal{L}(S_2) \Leftrightarrow w$ kodiert Belegung.

Wir zeigen später: $w \in \mathcal{L}(R_1) \cap \ldots \cap \mathcal{L}(R_k) \cap \mathcal{L}(S_1) \cap \mathcal{L}(S_2) \Leftrightarrow w$ kodiert Belegung, die Φ erfüllt. Dabei sind für $1 \leq i \leq k$ und $j \in \{1, 2\}$

$$\begin{aligned} R_i &= N^2 F_{i,1} F_{i,2} F_{i,3} N^2 \\ S_j &= U^2 W_j U^2 \end{aligned}$$

mit regulären Ausdrücken N, $F_{i,j}$, U und W_j aus $\mathrm{RE}(a, a^*)$. Der Ausdruck U beschreibt genau ein Wort u, d.h. $\mathcal{L}(U) = \{u\}$. N und U dienen als „Markierungen“, um die Komponenten von Φ unterscheiden zu können. Jedes $F_{i,j}$ in den R_i kodiert die Literale $L_{i,j}$ aus Φ.

Definiere W_1 und W_2 so, dass Wörter w in $W = W_1 \cap W_2$ Belegungen A_w für Φ sind und dass es für jede Belegung A ein Wort $w_A \in \mathcal{L}(W)$ mit $A_{w_A} = A$ gibt.

Wir konstruieren die Ausdrücke $R_1, \ldots, R_k, S_1$ und S_2, so dass gilt:

(i) $\{\epsilon, u\} \subseteq \mathcal{L}(N)$. Wenn N^l ein Wort $u^l z$ oder $z u^l$ matcht (für $l \in \mathbb{N}$), dann sei $z \in \mathcal{L}(\#^*)$.

(ii) Wenn $w \in \mathcal{L}(F_{i,j})$, dann macht A_w das Literal $L_{i,j}$ wahr. Wenn A das Literal $L_{i,j}$ wahr macht, dann ist $w_A \in \mathcal{L}(F_{i,j})$.

(iii) $u \in \mathcal{L}(F_{i,j})$

(iv) Jedes Wort in $\mathcal{L}(F_{i,j})$ beginnt und endet mit #, und # kommt sonst darin nicht vor.

(v) Jedes Wort in $\mathcal{L}(W)$ beginnt und endet mit #, und # kommt sonst darin nicht vor. $\mathcal{L}(W)$ enthält keine Wörter der Form $\#^*$.

Angenommen, es existieren Ausdrücke $R_1, \ldots, R_k, S_1$ und S_2, die die genannten Bedingungen erfüllen. Zeige nun: $\mathcal{L}(S_1) \cap \mathcal{L}(S_2) \cap \mathcal{L}(R_1) \cap \ldots \cap \mathcal{L}(R_k) \neq \varnothing \Leftrightarrow \Phi$ erfüllbar.

Zeige zunächst die Hinrichtung. Sei dazu $w \in \mathcal{L}(W)$ und $u^2 w u^2 \in \mathcal{L}(R_1) \cap \ldots \cap \mathcal{L}(R_k)$. Zeige nun, dass für alle $1 \leq i \leq k$ ein $j \in \{1, 2, 3\}$ existiert mit $w \in \mathcal{L}(F_{i,j})$.

Durch die # wird jedes Wort in N und F in Komponenten aufgeteilt. Wir nehmen an, dass ein i mit $1 \leq i \leq k$ existiert, so dass für alle $j \in \{1, 2, 3\}$ gilt: $w \notin \mathcal{L}(F_{i,j})$ und $u^2 w u^2 \in \mathcal{L}(R_i)$, d.h. $u^2 w u^2 \in \mathcal{L}(N^2 F_{i,1} F_{i,2} F_{i,3} N^2)$.

$$\underbrace{N \cdot N}_{u^2 w} \cdot \underbrace{F_{i,1} \cdot F_{i,1} \cdot F_{i,1} \cdot N \cdot N}_{u^2}$$
$$\underbrace{N \cdot N \cdot F_{i,1} \cdot F_{i,1} \cdot F_{i,1}}_{u^2} \cdot \underbrace{N \cdot N}_{w u^2}$$

Abbildung 1: Mögliche Lagen von $u^2 w u^2$ in R_i

Wegen der Aufteilung in Komponenten sind hier nur zwei Fälle möglich (vgl. Abbildung 1):

(a) $u^2 w \in \mathcal{L}(N^2) \Rightarrow w \in \mathcal{L}(\#^*)$ (Widerspruch zu (v))

(b) $w u^2 \in \mathcal{L}(N^2) \Rightarrow w \in \mathcal{L}(\#^*)$ (Widerspruch zu (v))

Also gibt es nach Bedingung (ii) für alle $1 \leq i \leq k$ ein $j \in \{1,2,3\}$, so dass A_w ein Literal $L_{i,j}$ erfüllt. Da es sich um eine Formel in konjunktiver Normalform handelt, erfüllt die Belegung A_w auch jedes C_i und damit auch Φ.

Für die Rückrichtung nehmen wir an, dass Φ durch eine Belegung A erfüllt wird. Da Φ in konjunktiver Normalform ist, gibt es für alle $1 \leq i \leq k$ ein $j_i \in \{1,2,3\}$, so dass L_{i,j_i} von A erfüllt wird. Mit Bedingung (ii) folgt, dass für alle $1 \leq i \leq k$ ein $j_i \in \{1,2,3\}$ existiert mit $w_A \in \mathcal{L}(F_{i,j})$ und $w_A \in \mathcal{L}(W) = \mathcal{L}(W_1) \cap \mathcal{L}(W_2)$. Daher gilt: $u^2 w_A u^2 \in S_1$ und $u^2 w_A u^2 \in S_2$.

Zu zeigen ist, dass $u^2 w_A u^2 \in \mathcal{L}(R_i)$ für alle $1 \leq i \leq k$. Es gilt:

(1) $u \in \mathcal{L}(N)$ (nach Bedingung (i))

(2) $u \in \mathcal{L}(F_{i,j})$ für $j \in \{1,2,3\}$ (nach Bedingung (iii))

(3) $\epsilon \in \mathcal{L}(N)$ (nach Bedingung (i))

Prüfe für $j \in \{1,2,3\}$, ob $u^2 w_A u^2$ in $\mathcal{L}(R_i) = \mathcal{L}(N^2 F_{i,1} F_{i,2} F_{i,3} N^2)$ liegt:

	N	$\cdot$	N	$\cdot$	$F_{i,1}$	$\cdot$	$F_{i,2}$	$\cdot$	$F_{i,3}$	$\cdot$	N	$\cdot$	N
$j=1$:	u	$\cdot$	u	$\cdot$	w_A	$\cdot$	u	$\cdot$	u	$\cdot$	ϵ	$\cdot$	ϵ
$j=2$:	ϵ	$\cdot$	u	$\cdot$	u	$\cdot$	w_A	$\cdot$	u	$\cdot$	u	$\cdot$	ϵ
$j=3$:	ϵ	$\cdot$	ϵ	$\cdot$	u	$\cdot$	u	$\cdot$	w_A	$\cdot$	u	$\cdot$	u

Also liegt $u^2 w_A u^2$ in allen $\mathcal{L}(R_i)$ und die Behauptung wurde gezeigt.

Nun müssen Ausdrücke $R_1, \ldots, R_k, S_1$ und S_2, die die oben genannten fünf Bedingungen erfüllen, konstruiert werden. Die regulären Ausdrücke N haben die Form $\#^* a^* \$^* a^* \$^* \ldots \$^* a^* \#^*$ (n-mal a^*) und U sei definiert als $\#a\$a\$\ldots\$a\#$ (n-mal a).

Wir definieren die Ausdrücke r_0, r_1, r_*, s_{01}, s'_{01} und r_{01} wie folgt: $r_0 = b^* a^+$, $r_1 = a^+ b^*$, $r_* = a^* b^* a^*$, $s_{01} = b^* a^+ b^*$, $s'_{01} = a^* b^+ a^*$, $r_{01} = s_{01} \cap s'_{01} = a^+ b^+ + b^+ a^+$. Die Beziehungen dieser Ausdrücke zueinander sind in Abbildung 2 dargestellt.

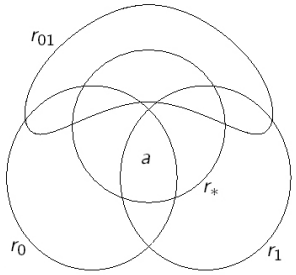

Abbildung 2: Beziehungen der regulären Ausdrücke r_0, r_1, r_* und r_{01}

Sie erfüllen die folgenden Bedingungen:

$$a \in \mathcal{L}(r_0) \cap \mathcal{L}(r_1) \cap \mathcal{L}(r_*) \tag{INT1}$$
$$\mathcal{L}(r_0) \cap \mathcal{L}(r_{01}) \cap \mathcal{L}(r_*) \neq \varnothing \tag{INT2}$$
$$\mathcal{L}(r_1) \cap \mathcal{L}(r_{01}) \cap \mathcal{L}(r_*) \neq \varnothing \tag{INT3}$$
$$\mathcal{L}(r_0) \cap \mathcal{L}(r_1) \cap \mathcal{L}(r_{01}) = \varnothing \tag{INT4}$$

Zeugen für (INT2) und (INT3) sind $z_0 = ba$ bzw. $z_1 = ab$. Der Schnitt $\mathcal{L}(r_0) \cap \mathcal{L}(r_1) \cap \mathcal{L}(r_{01})$ ist leer (INT4), weil Wörter in $\mathcal{L}(r_0)$ immer mit einem a enden müssen und Wörter in $\mathcal{L}(r_1)$ immer mit einem a beginnen müssen, aber Wörter in $\mathcal{L}(r_{01})$ von der Form a^+b^+ oder b^+a^+ sind.

Weiterhin sei $W_1 = \#s_{01}\$\ldots\$s_{01}\#$ (n-mal s_{01}) und $W_2 = \#s'_{01}\$\ldots\$s'_{01}\#$ (n-mal s'_{01}) und $W = W_1 \cap W_2$. Mit dem Wort $w = \#w_1\$\ldots\$w_n\#$ sei die Belegung A_w verknüpft: $A_w(x_j) = \text{true}$, wenn $w_j \in L(r_1)$, und $A_w(x_j) = \text{false}$, sonst. Seien $z_0 \in \mathcal{L}(r_0) \cap \mathcal{L}(r_{01}) \cap \mathcal{L}(r_*)$ und $z_1 \in \mathcal{L}(r_1) \cap \mathcal{L}(r_{01}) \cap \mathcal{L}(r_*)$, die wegen (INT2) und (INT3) existieren (vgl. auch Abbildung 3). Nach Bedingung (INT4) gilt: $z_0 \notin \mathcal{L}(r_1)$ und $z_1 \notin \mathcal{L}(r_0)$ (vgl. auch Zeugen für (INT2) und (INT3).

Für eine Belegung A sei w_A das Wort $\#y_1\$\ldots\$y_n\#$, mit $y_j = z_1$, wenn $A(x_j) = \text{true}$, und $y_j = z_0$, sonst. Setze weiterhin für alle i: $F_{i,j} = \#e_1\$\ldots\$e_n\#$, mit $e_l = r_0$, wenn $L_{i,j} = \neg x_l$, $e_l = r_1$, wenn $L_{i,j} = x_l$, und $e_l = r_*$, sonst.

Nun muss noch gezeigt werden, dass die Bedingungen (i) bis (v) von den konstruierten Ausdrücken $R_1, \ldots, R_k, S_1$ und S_2 erfüllt werden. Für die Bedingung (i) folgt

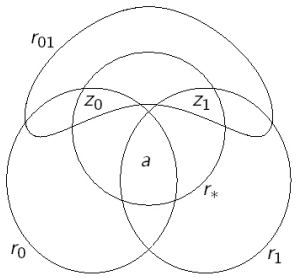

Abbildung 3: Lage von z_0 und z_1 innerhalb der regulären Ausdrücke r_0, r_1, r_* und r_{01}

dies unmittelbar aus der Definition von N. Bedingung (iii) gilt wegen (INT1) und der Definition von U. Die Bedingungen (iv) und (v) gelten wegen der Definition von $F_{i,j}$ und W.

Für Bedingung (ii) sei w ein Wort, das von $F_{i,j} = \#e_1\$ \ldots \$e_n\#$ gematcht wird. Wenn $L_{i,j} = x_l$, dann ist $e_l = r_1$ und $w_l \in \mathcal{L}(r_1)$. Daher gilt: $A_w(x_l) = \text{true}$. Analog für $L_{i,j} = \neg x_l$. Sei nun $L_{i,j} = x_l$ und $F_{i,j} = \#e_1\$ \ldots \$e_n\#$. Nach Definition ist $e_l = r_1$ und für alle anderen e_l gilt: $e_l = r_*$. Wenn A $L_{i,j}$ wahr macht und $w_A = \#y_1\$ \ldots \$y_n\#$, dann ist $y_l = z_1$ und alle anderen y_l sind aus $\{z_0, z_1\}$. Da $z_1 \in \mathcal{L}(r_1) \cap \mathcal{L}(r_*)$ und $\{z_0, z_1\} \subseteq \mathcal{L}(r_*)$, folgt: $w_A \in F_{i,j}$. Analog für $L_{i,j} = \neg x_l$.

Für die NP-Vollständigkeit des Schnittproblems für $\text{RE}(a, a^*)$ betrachte die Ausdrücke $r_1, \ldots, r_n$ aus dieser Klasse von einfachen regulären Ausdrücken. Rate für jedes r_i und jedes a^* in r_i, ob a^* entweder durch das leere Wort oder durch a^+ interpretiert wird. Dadurch ändert sich an der Ausdrucksstärke der regulären Ausdrücke nichts. Betrachte daher die Ausdrücke $r'_1, \ldots, r'_n$ aus $\text{RE}(a, a^+)$. Da nach Theorem 3 das Schnittproblem für $\text{RE}(a, a^+)$ in PTIME entscheidbar ist, folgt, dass das Schnittproblem für $\text{RE}(a, a^*)$ in NP liegt. ■

4 Zusammenfassung

In dieser Arbeit wurden zunächst reguläre Ausdrücke, Dokumenttypdefinitionen und Entscheidungsprobleme vorgestellt.

In den beiden folgenden Abschnitten wurde bewiesen, dass es sowohl effizient als auch ineffizient entscheidbare Entscheidungsprobleme von Klassen von einfachen regulären Ausdrücken gibt. Zu den effizient entscheidbaren Problemen gehören das Äquivalenzproblem für $\text{RE}(a, a^*)$ (vgl. Theorem 1) und das Schnittproblem für $\text{RE}(a, a^+)$ (vgl. Theorem 3), die jeweils in PTIME liegen. Das Schnittproblem für $\text{RE}(a, a^*)$ (vgl. Theorem 2) ist NP-vollständig und somit nicht effizient entscheidbar.

A Literatur

[BM04] BEHME, Henning ; MINTERT, Stefan.
XML in der Praxis, Extensible Markup Language für Profis.
http://www.linkwerk.com/pub/xmlidp/2000/.
2004

[HMU02] HOPCROFT, John E. ; MOTWANI, Rajeev ; ULLMAN, Jeffrey D.:
Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie.
2. Auflage.
Pearson Studium, 2002. –
Kap. 5.3.4

[Koz73] KOZEN, Dexter:
Lower bounds for natural proof systems.

In: *Proceedings 18th Annual Symposium on Foundations of Computer Science*, IEEE, 1973, S. 254–266

[MNS04] MARTENS, Wim ; NEVEN, Frank ; SCHWENTICK, Thomas:
Complexity of Decision Problems for Simple Regular Expressions.
In: *29th Symposium on Mathematical Foundations of Computer Science (MFCS 2004)*, 2004, S. 889–900

[Sch01] SCHÖNING, Uwe:
Theoretische Informatik - kurzgefasst.
4. Auflage.
Spektrum, Akad. Verl., 2001

[SEL05] SELFHTML E.V.
XML/DTDs.
http://de.selfhtml.org/xml/.
2005

[SM73] STOCKMEYER, L. J. ; MEYER, A. R.:
Word problems requiring exponential time (Preliminary Report).
In: *STOC '73: Proceedings of the fifth annual ACM symposium on Theory of computing*, ACM Press, 1973, S. 1–9

[Wik05a] WIKIMEDIA FOUNDATION INC.
Dokumenttypdefinition.
http://de.wikipedia.org/wiki/DTD.
2005

[Wik05b] WIKIMEDIA FOUNDATION INC.
Erfüllbarkeitsproblem der Aussagenlogik.
http://de.wikipedia.org/wiki/Erf%C3%BCllbarkeitsproblem_der_Aussagenlogik.
2005