Reguläre Ausdrücke

Katja Markert

Institut für Computerlinguistik
Uni Heidelberg
markert@cl.uni-heidelberg.de
(Zum Teil mit Folien von Anette Frank oder Yannick Versley)

October 29, 2019

Heute

1. Spezifikation von Zeichenketten
2. Reguläre Ausdrücke
3. zum Matching
4. zur Spezifikation

Motivation

- Suche nach Wörtern: verschiedene Wortformen

  Student
  Studentin
  Studenten
  Studentinnen
  ...

  Mind. 4 verschiedene Wortformen $\longrightarrow$ 4 Suchoperationen?
  Besser: Suche nach *Student*($|(i|e|inne)n$)
- Anwendungen: Suche, Rechtschreibkorrektur, Ersetzungen, Eigennamenerkennung ...

Terminologie

- **Alphabet**: Eine (endliche, nicht-leere) Menge von Zeichen/Symbolen. Beispiel:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{a, 1, 4, \%, \tilde{O}, \lambda, \ddot{\smile}\}$$

- **Zeichenketten (Strings)**: konkatenierte Zeichen

$$\Sigma_2^* \supseteq \{\varepsilon, aa, 1a4a, 14\%, \lambda\lambda, \lambda, \ldots\}$$

- ε empty string
- **Formale Sprache L über Alphabet Σ**: Menge von Zeichenketten, beliebige Teilmenge von Σ^*

$$L_2 = \{\lambda, \lambda\lambda, \lambda\lambda\lambda, \lambda\lambda\lambda\lambda, \ldots\} \subseteq \Sigma_2^*$$

Terminologie

- Konkatenation von Zeichenketten $\circ$:

$$\lambda\lambda \circ 14\% = \lambda\lambda 14\%$$

$$\lambda\lambda \circ \varepsilon = \lambda\lambda$$

- Operationen auf Sprachen (Beispiele mit L_2 von vorheriger Folie):

$$L_2 \circ \{14\%\} = \{\lambda 14\%, \lambda\lambda 14\%, \lambda\lambda\lambda 14\%, \ldots\}$$

$$L_2 \circ \emptyset = \emptyset$$

$$L_2 \circ \{\varepsilon\} = L_2$$

$$L_2 \cup \{14\%\} = \{14\%, \lambda, \lambda\lambda, \lambda\lambda\lambda, \ldots\}$$

$$L_2 \cup \emptyset = L_2$$

Reguläre Ausdrücke

- Ein regulärer Ausdruck spezifiziert/ist eine Menge von Zeichenketten
- Zur Definition einer Sprache (= Menge von Zeichenketten)
- Zur Suche nach Mustern in einem Text (Korpus)

Definition

Definition regulärer Ausdrücke (induktiv)

Menge Reg_Σ der regulären Ausdrücke über einem gegebenen endlichen Alphabet Σ:

1. Konstanten: $\emptyset \in Reg_\Sigma$; $\varepsilon := \{\varepsilon\} \in Reg_\Sigma$; $a(:= \{a\}) \in Reg_\Sigma \forall a \in \Sigma$
2. Operationen: Falls $R, S \in Reg_\Sigma$, dann sind die folgenden Mengen auch in Reg_Σ:
   - Konkatenation aus einem String aus R und einem in S: RS
   - Veroderung= Vereinigung von R und S: $R|S$
   - Kleene Star: R^* bzw $R*$ kleinste Übermenge von R, die ε enthält sowie unter Konkatenation abgeschlossen ist.

Schreibweise: $R|S$ oft als $R + S$ oder $R \cup S$ geschrieben.

Beispiele

Oft zur Abgrenzung in Slashes.

Über dem Ascii Alphabet:

- /*Medici*/
- /*Opfer Noahs*/
- /*Opfer noahs*/
- /*i*/
- /(*j*|*J*)*unge Frau*/ oder /[*jJ*]*unge Frau*/
- /1|2|3|4|5|6|7|8|9|0/ oder /[1234567890]/
- /a^*/
- /$(ab)^*$/
- /$ab(ab)^*$/
- Äquivalent: $(a|b)^*$ und $(a^*b^*)^*$

Extended Regular Expressions

Definition der regulären Ausdrücke ist minimal. Zur Abkürzung werden oft noch andere Operationen und Zeichenklassen eingeführt, die aber die Expressivität nicht verändern.

Beispiele auf den folgenden Folien.

Optionalität und Wiederholung

- Kleene $*$: keine oder beliebig viele Instanzen. Beispiel: a^*
- Kleene $+$: eine oder beliebig viele Instanzen. Beispiel: $a+$ ($= aa^*$). Was matched $1+0$ über dem Alphabet $\{1,0\}$?
- Optionalität ?: eine oder keine Instanz. Beispiele:
  - $a?$ ($= a|\varepsilon$)
  - $/colou?r/$
- $\{m,n\}$: zwischen m und n Instanzen. Beispiel: $/Ba\{2,4\}!/$
- $\{n\}$: genau n Instanzen. Beispiel: $/Ba\{3\}!/$

Ranges, Zeichenklassen, Wildcard

Hängt oft von spezifischen Standards und Programmiersprachen ab:

Klasse/Muster	Matches	Beispiel
$[A-Z]$	Großer Buchstabe	**D**renched Blossoms
$[a-z]$	Kleiner Buchstabe	**m**y beans were impatient
$[A-Za-z]$	Buchstabe	4 **D**renched Blossoms
$[0-9]$	Einzelne Ziffer	Chapter **1**: Down the Rabbit hole
$\backslash d$	Einzelne Ziffer	Chapter **1**: Down the Rabbit hole
. (Punkt)	Jedes Zeichen aus Σ	

Negation

Muster	Matches	Beispiel
[^A – Z]	Kein großer Buchstabe	D**r**enched Blossoms
[^S s]	Weder *s* noch *S*	S**o**phie was impatient
[e^]	*e* oder ^	Chapt**e**r 1
a^b	Das Muster a^b	Look up a^b now

Negation ^ direkt nach eckiger Klammer

Anker, Sonderzeichen und geschützte Zeichen

- Begrenzer (Anker): Beginn/Ende einer Zeile.
  - ^: Kennzeichnet den Beginn einer Zeile
  - $: Kennzeichnet Ende einer Zeile
  - /^*The*.**dog*\.$/ spezifiziert alle Zeilen, die mit *The* beginnen und mit *dog.* enden
- Zeilenumbruch: \n
- Leerzeichen: \s \t
- geschützte Zeichen: \. * \[\] \\ \+ \? \| \(\) \$ usw.

Noch ein paar Beispiele

Muster	Beispiele
colou?r	**color colour**
oo**h*!	**oh! ooh! oooh!** . . .
o$^+$*h*!	**oh! ooh! oooh!** . . .
beg.**n*	**begin begun beg345tn**

Präzedenz der Operatoren

1. Klammern ()
2. Zähler *, +, ?, {n}, {n,m}
3. Sequenzen und Anker *abc* ^ $
4. Disjunktion |

Beispiele:

- /*fein*|*einer*|*einsten*/ $\rightarrow$ *fein,einer,einsten*
- /*f*(*ein*|*einer*|*einsten*)/ $\rightarrow$ *fein,feiner,feinsten*
- /(*Eur* [0 – 9]$^{+}$,)$^{+}$/ $\rightarrow$ EUR 13, EUR 22, EUR 99, ...
- /*the**/: *theeeee*
- /[*a* – *z*]*/: greedy matching

Regexpal

Try out regular expression matching at

`https://www.regexpal.com/`

Welche Zeichenketten werden erkannt?

- $/a^*(b|cd)^+d?a\{3\}/$
  1. cdbaaa
  2. aadaaa
  3. abcdcdaaa
- $/[\hat{}\ a-z][a-z]^*[1-3][0-9]\{3\}/$
  1. Inf325
  2. Dkfz2034
  3. 63865

Welche Zeichenketten werden erkannt?

- $/a^*(b|cd)^+d?a\{3\}/$
  1. cdbaaa wird erkannt
  2. aadaaa wird nicht erkannt
  3. abcdcdaaa wird erkannt
- $/[\hat{}\ a-z][a-z]^*[1-3][0-9]\{3\}/$
  1. Inf325 wird nicht erkannt
  2. Dkfz2034 wird erkannt
  3. 63865 wird erkannt

Schreiben Sie reguläre Ausdrücke

Gegeben sei jeweils das Alphabet $\Sigma = \{a, b\}$. Schreiben Sie jeweils einen regulären Ausdruck für die folgenden Mengen von Zeichenketten.

- Alle Zeichenketten, die nicht mit *ab* enden. Ohne Verwendung des Negationscharakters.
- Alle Zeichenketten gerader Länge.
- Alle Zeichenketten, die genausoviele *a*s wie *b*s enthalten

Suche mit regulären Ausdrücken

Wie einfach ist dies? Suchen wir nach Vorkommen des englischen definiten Artikels *the*.

- /*the*/
- /[*Th*]*he*/
- / [*Tt*]*he*/
- /(|,|\.)[*Th*]*he*(|,|\.)/
- /[^*a*−*zA*−*Z*][*Tt*]*he*[^*a*−*zA*−*Z*]/
- /(^|[^*a*−*zA*−*Z*])[*Tt*]*he*([^*a*−*zA*−*Z*]|$)/

Evaluation von Suchakkuratheit

Recall

Vermeide unvollständige Suchergebnisse. Wieviele der wirklichen Vorkommen habe ich gefunden?

Precision

Vermeide falsche Suchresultate. Von meinen Suchergebnissen, wieviele waren korrekt?

Konfusionsmatrix und Precision/Recall Formeln

		Predicted True	Predicted False
Known	True	TP	FN
Known	False	FP	TN

True Positives, False Positives, True Negatives, False Negatives

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - measure = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Beispiel

- Suche nach Vorkommen von *the* (Artikel) in einem Text mit 50 Wörtern
- Predicted: *the* (5mal), *there* (2mal), *Theology* (einmal), *The* (zweimal)
- Truth: *the* 5mal, *The* zweimal

Berechnen Sie Konfusionsmatrix, Precision, Recall, F-measure, Accuracy

Ersetzen mit regulären Ausdrücken

- Suche und Substitution in Commandline oder vielen Progsprachen

$$s/colour/color/$$

- Markierung von Teilsequenzen oft möglich. Nicht Teil normaler regulärer Sprachen (da Gedächtnis notwendig)
  - EUR 100 $\rightarrow$ 100 EUR
  - $s/EUR\ ^{+}([0-9]^{+})/\backslash 1\ EUR/$

Fazit

- Reguläre Ausdrücke spezifizieren Mengen von Zeichenketten (= formale Sprache)
- Sie können zur Suche nach Mustern oder Wortformen in Korpora eingesetzt werden
- Vielzahl von Operatoren und Metacharakteren
- Wichtige Begriffe: Operatorenpräzedenz, Konfusionsmatrix, Precision, Recall, F-measure, Accuracy

Literatur und Materialien

- Übungsblatt 2
- Jurafsky und Martin (3rd edition) Kapitel 2
- Hopcroft, Motwani und Ullman (2013): Introduction to Automata Theorey, Languages and Computation (3rd edition). Pearson
- Knapp, Stefan (2018): Übungsbuch Automaten und Formale Sprachen. 117 Aufgaben und Lösungen. Springer. E-book verlinkt von der Kurswebseite oder siehe Heidi.