Reguläre Sprachen und endliche Automaten

Motivation: Syntaxüberprüfung

Definition: Fließkommazahlen in Java

> A floating-point literal has the following parts: a whole-number part, a decimal point (represented by an ASCII period character), a fractional part, an exponent, and a type suffix. The exponent, if present, is indicated by the ASCII letter e or E followed by an optionally signed integer.
>
> At least one digit, in either the whole number or the fraction part, and either a decimal point, an exponent, or a float type suffix are required. All other parts are optional.
>
> A floating-point literal is of type float if it is suffixed with an ASCII letter F or f; otherwise its type is double and it can optionally be suffixed with an ASCII letter D or d.

Motivation: Syntaxüberprüfung

Eine derartige Überprüfung von Hand zu implementieren ist höchst langwierig und fehleranfällig.

Ähnliche Probleme finden sich nicht nur beim Compilerbau, sondern überall, wo Eingaben bestimmte Formatvorschriften erfüllen müssen.

Z. B.: Mailadressen, Zahlformate, Datenbankabfragen.

Frage: Kann man sich die Von-Hand-Implementierung sparen?
Könnte ein Programm die Implementierung automatisch erzeugen, wenn man ihm die Formatvorschrift in „lesbarer“ Form eingibt?

Theoretische Grundbegriffe

Ein Alphabet A ist eine Menge von Zeichen/Symbolen (hier: endlich).

Ein Wort (String) über einem Alphabet A ist eine Folge $x_1 x_2 \ldots x_n$, wobei $n \geq 0$ ist und jedes x_i ein Zeichen aus A ist.

Notationen:

$A^* =$ Menge aller Wörter über A („Kleene star“).

$\varepsilon \;\; =$ leeres Wort

$a^n =$ Abkürzung für Wiederholungen eines Zeichens ($a^0 = \varepsilon$, $a^1 = a$, $a^2 = aa$, und so weiter)

Eine Sprache ist eine Teilmenge von A^*.

Theoretische Grundbegriffe

Beispiele für Sprachen

Leere Menge.

Menge aller Wörter über A: A^*.

Menge, die nur das Wort „$ccac$“ enthält.

Menge aller Wörter der Länge 3 über A.

Menge aller Wörter über A, die ein „c“ enthalten.

Menge aller Wörter über A, die kein „c“ enthalten.

Theoretische Grundbegriffe

Beispiele für Sprachen

Menge aller Wörter, die aus einer Folge von „a“s
und dahinter einer Folge von „b“s bestehen:
$\{ a^n b^m \mid n \geq 0, m \geq 0 \}$.

Menge aller Wörter, die aus einer Folge von „a“s
und dahinter einer gleichlangen Folge von „b“s bestehen:
$\{ a^n b^n \mid n \geq 0 \}$.

Menge aller Java-Programme.

Menge aller stets terminierenden Java-Programme.

Deterministische endliche Automaten

Beispiel: Sei L die Menge aller Wörter über $\{a, b, c\}$,
die genau ein „a“ enthalten.

Wie kann man entscheiden, ob ein Wort zur Sprache L gehört?

Wir lesen die Zeichen des Wortes von links nach rechts
und merken uns, ob noch kein, ob genau ein,
oder ob mehr als ein „a“ gefunden wurde.

Deterministische endliche Automaten

Wir befinden uns also innerhalb des Programms jeweils in einem von drei Zuständen:

$0 =$ noch kein „a“ gefunden.

$1 =$ genau ein „a“ gefunden.

$2 =$ mehr als ein „a“ gefunden

Wenn im Zustand 0 ein „a“ gelesen wird, gehen wir in den Zustand 1, wenn ein anderes Zeichen gelesen wird, bleiben wir im Zustand 0.

Wenn im Zustand 1 ein „a“ gelesen wird, gehen wir in den Zustand 2, wenn ein anderes Zeichen gelesen wird, bleiben wir im Zustand 1.

Wenn im Zustand 2 ein beliebiges Zeichen gelesen wird, bleiben wir im Zustand 2.

Wenn wir zum Schluß im Zustand 1 sind, gehört das gelesene Wort zur Sprache L.

Deterministische endliche Automaten

Die Zustandsübergänge kann man tabellarisch darstellen:

	a	b	c
0	1	0	0
1	2	1	1
2	2	2	2

Oft verwendet man auch eine graphische Darstellung:

Zustände $\rightsquigarrow$ Knoten eines Graphs,

Übergänge $\rightsquigarrow$ beschriftete Kanten.

Deterministische endliche Automaten

Ein *deterministischer endlicher Automat* (DEA) über einem Alphabet A besteht aus:

- einer *endlichen* Menge von Zuständen Q,
- einem Anfangszustand $q^0 \in Q$,
- einer Menge von Endzuständen $Q^E \subseteq Q$,
- einer Übergangsfunktion $\delta : Q \times A \to Q$.

Deterministische endliche Automaten

Ein DEA beginnt im Anfangszustand q^0.

Er liest die Zeichen des Wortes sequentiell von vorn nach hinten.

Wenn sich der Automat in einem Zustand q befindet und ein Zeichen x liest, geht er in den Zustand $\delta(q, x)$ über.

Wenn er sich am Wortende in einem Zustand aus Q^E befindet, *akzeptiert* er das Wort.

Die Menge aller Wörter, die ein Automat akzeptiert,
ist die *von diesem Automaten akzeptierte Sprache*.

Deterministische endliche Automaten

Beispiel: $A = \{a, b, c\}$, $L = \{ab, ba\}$.

δ	a	b	c
0	1	2	4
1	4	3	4
2	3	4	4
3	4	4	4
4	4	4	4

$q^0 = 0$.

$Q^E = \{3\}$.

Deterministische endliche Automaten

Beispiel: $A = \{a, b, c\}$,

$L =$ Menge aller Wörter, in denen das Teilwort *ab* oder *ba* vorkommt.

δ	a	b	c
0	1	2	0
1	1	3	0
2	3	2	0
3	3	3	3

$q^0 = 0$.

$Q^E = \{3\}$.

Deterministische endliche Automaten

Beispiel: $A = \{a, b, c\}$, $L = \{\, a^n b^n \mid n \geq 0 \,\}$.

Für diese Sprache existiert kein DEA, der sie akzeptiert.

Beweis: Angenommen, es gäbe einen DEA.
Da dieser nur endlich viele Zustände hat, muß es verschiedene Zahlen i und k geben, so daß der Automat vom Anfangszustand aus sowohl nach dem Lesen von a^i als auch nach dem Lesen von a^k im gleichen Zustand q ist.
Der Automat akzeptiert die Sprache L; da $a^i b^i$ in L enthalten ist, muß er also einen Endzustand erreichen, wenn er von q aus b^i liest.
Dann akzeptiert er aber auch das Wort $a^k b^i$, und dieses Wort ist nicht in L enthalten.

Intuitiv: DEAs können nicht beliebig weit zählen.

Reguläre Sprachen

Eine Sprache L heißt regulär, wenn es einen DEA gibt, der L akzeptiert.

Reguläre Sprachen

Reguläre Sprachen sind unter vielen Operationen abgeschlossen:

Wenn L eine reguläre Sprache ist, dann ist auch das Komplement von L, also die Sprache $A^* \setminus L = \{ w \in A^* \mid w \notin L \}$ regulär:

Beweis: Nimm den DEA, der L akzeptiert, ersetze Q^E durch $Q \setminus Q^E = \{ q \in Q \mid q \notin Q^E \}$.

Reguläre Sprachen

Wenn L_1 und L_2 reguläre Sprachen sind, dann ist auch der Durchschnitt $L_1 \cap L_2$ regulär.

Beweis: Seien Z_1 und Z_2 Automaten, die L_1 bzw. L_2 akzeptieren.
$Z_1 = (Q_1, q_1^0, Q_1^E, \delta_1)$.
$Z_2 = (Q_2, q_2^0, Q_2^E, \delta_2)$.

Dann konstruieren wir einen neuen DEA $Z = (Q, q^0, Q^E, \delta)$:

$Q = \{\, (q_1, q_2) \mid q_1 \in Q_1,\ q_2 \in Q_2 \,\}$.

$q^0 = (q_1^0, q_2^0)$.

$Q^E = \{\, (q_1, q_2) \mid q_1 \in Q_1^E,\ q_2 \in Q_2^E \,\}$.

$\delta((q_1, q_2), x) = (\delta(q_1, x), \delta(q_2, x))$.

Reguläre Sprachen

Auch die Vereinigung $L_1 \cup L_2$ ist regulär.

Beweis: Gleiche Konstruktion wie beim Durchschnitt, außer:

$Q^E = \{\, (q_1, q_2) \mid q_1 \in Q_1^E \;\text{ oder }\; q_2 \in Q_2^E \,\}.$

Nichtdeterministische endliche Automaten

Bisher: Automat geht beim Lesen eines Zeichens x von einem Zustand q in einen definierten Folgezustand $\delta(q, x)$ über.

Man kann endliche Automaten so verallgemeinern, daß man auch mehrere oder gar keine Folgezustände zuläßt:
Statt einer Übergangsfunktion bekommen wir eine Übergangsrelation.

Außerdem kann man auch zulassen, daß der Automat ohne ein Zeichen zu lesen von einem Zustand in einen anderen wechselt (ε-Übergänge).

Man nennt so einen Automaten einen *nichtdeterministischen endlichen Automaten* (NEA).

Jeder DEA ist auch ein NEA, aber nicht umgekehrt.

Nichtdeterministische endliche Automaten

Frage: was akzeptiert ein NEA ?
Wenn man vom Startzustand aus beim Lesen des Zeichens „a“ die Zustände 1 und 2 erreichen kann, wobei 1 ein Endzustand ist aber 2 kein Endzustand ist, wird dann das Wort „a“ akzeptiert?

Definition: Ein NEA akzeptiert ein Wort, wenn es mindestens eine Möglichkeit gibt, das Wort *vollständig* zu lesen und dabei vom Anfangs- in einen Endzustand zu gelangen.

Dabei gilt, daß in einem Zustand q ein Zeichen x nur dann gelesen werden kann, wenn für die Kombination (q, x) mindestens ein Folgezustand definiert ist.

Nichtdeterministische endliche Automaten

Viele Sprachen lassen sich mittels eines NEA leichter beschreiben als mit einem DEA
(leichter = intuitiver und/oder mit weniger Zuständen).

Beispiel: L = Menge aller Wörter über $\{a, b, c\}$, bei denen das letzte Zeichen vorher schon einmal im Wort vorgekommen ist.

Nichtdeterministische endliche Automaten

Aber: Für jede Sprache, die von einem NEA akzeptiert wird, existiert auch ein DEA, der sie akzeptiert.

Beweisidee: Potenzmengenkonstruktion

Nehmen wir an, wir haben einen NEA N, der die Sprache L akzeptiert.
Wir konstruieren einen DEA D:

Wenn N die Zustandsmenge Q hat, dann ist jede Teilmenge von Q ein Zustand von D.

Beispiel:
Zustände von N: 0, 1, 2.
Zustände von D: $\emptyset$, $\{0\}$, $\{1\}$, $\{2\}$, $\{0,1\}$, $\{0,2\}$, $\{1,2\}$, $\{0,1,2\}$.

Nichtdeterministische endliche Automaten

Übergänge von D:

Beispiel: N geht beim Lesen von „a“
im Zustand 0 in den Zustand 1 oder 2 über,
im Zustand 1 in den Zustand 0 über,
im Zustand 2 nirgendwohin.

Dann kann D beim Lesen von „a“ folgende Übergänge ausführen:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\emptyset & \mapsto & \emptyset & \{0,1\} & \mapsto & \{0,1,2\} \\
\{0\} & \mapsto & \{1,2\} & \{0,2\} & \mapsto & \{1,2\} \\
\{1\} & \mapsto & \{0\} & \{1,2\} & \mapsto & \{0\} \\
\{2\} & \mapsto & \emptyset & \{0,1,2\} & \mapsto & \{0,1,2\}
\end{array}$$

Nichtdeterministische endliche Automaten

Anfangszustand von D ist die Menge, die aus dem Anfangszustand von N besteht.

Endzustände von D sind alle Mengen, die Endzustände von N enthalten.

Dann gilt: D akzeptiert genau die Wörter, die N akzeptiert.

Nichtdeterministische endliche Automaten

Die folgenden Eigenschaften sind äquivalent:

Die Sprache L ist regulär.

Es existiert ein DEA D, der L akzeptiert.

Es existiert ein NEA N, der L akzeptiert.

Problem: Die Zustandsmenge von D ist möglicherweise exponentiell größer als die Zustandsmenge von N.

Nichtdeterministische endliche Automaten

Folgerung: Wenn L_1 und L_2 reguläre Sprachen sind, dann ist auch die Konkatenation $L_1 \cdot L_2 = \{ w_1 w_2 \mid w_1 \in L_1,\ w_2 \in L_2 \}$ regulär.

Beweis: Verbinde jeden Endzustand des NEA, der L_1 akzeptiert, mittels eines ε-Übergangs mit dem Anfangszustand des NEA, der L_2 akzeptiert.

Nichtdeterministische endliche Automaten

Folgerung: Wenn L eine reguläre Sprache ist, dann ist auch der nichtleere Abschluß $L^+ = \{ w_1 w_2 \ldots w_n \mid n > 0,\ w_i \in L \}$ regulär.

Beweis: Verbinde jeden Endzustand des NEA, der L akzeptiert, mittels eines ε-Übergangs mit dem Anfangszustand des Automaten.

Nichtdeterministische endliche Automaten

Folgerung: Wenn L eine reguläre Sprache ist, dann ist auch der Abschluß $L^* = L^+ \cup \{\varepsilon\}$ regulär.

Reguläre Ausdrücke

Sei A ein Alphabet.

Wir bezeichnen $\emptyset$, $\{\varepsilon\}$ und $\{x\}$ (für jedes $x \in A$) als Grundsprachen.

Wir bezeichnen die Operationen Vereinigung ($L_1 \cup L_2$), Konkatenation ($L_1 \cdot L_2$) und Abschluß (L^*) als reguläre Operationen.

Satz: Eine Sprache ist genau dann regulär, wenn sie sich aus den Grundsprachen mittels der regulären Operationen aufbauen läßt.

Beweis:
($\Leftarrow$) Die Grundsprachen sind regulär, die regulären Operationen erhalten die Regularität.

($\Rightarrow$) Sehr technisch.

Reguläre Ausdrücke

Reguläre Ausdrücke beschreiben den Aufbau einer Sprache aus Grundsprachen mittels regulärer Operationen:

Das Symbol $\emptyset$ ist ein regulärer Ausdruck, der die leere Menge $\emptyset$ beschreibt.

Das Symbol ε ist ein regulärer Ausdruck, der die Sprache $\{\varepsilon\}$ beschreibt, die nur aus dem leeren Wort besteht.

Wenn x ein Zeichen aus A ist, dann ist x ein regulärer Ausdruck, der die Sprache $\{x\}$ beschreibt, die nur aus dem Wort x besteht.

Reguläre Ausdrücke

Wenn r_1 und r_2 reguläre Ausdrücke sind, die die Sprachen L_1 und L_2 beschreiben,
dann ist $r_1 r_2$ ein regulärer Ausdruck, der die Sprache $L_1 \cdot L_2$ beschreibt,
und $r_1 | r_2$ ist ein regulärer Ausdruck, der die Sprache $L_1 \cup L_2$ beschreibt.

Wenn r ein regulärer Ausdruck ist, der die Sprache L beschreibt,
dann ist r^* ein regulärer Ausdruck, der die Sprache L^* beschreibt.

Zusätzlich: Klammern nach Bedarf.

Reguläre Ausdrücke

Beispiele:

Der reguläre Ausdruck $a(b|c)$ beschreibt die Sprache $\{ab, ac\}$.
Auch der reguläre Ausdruck $ab|ac$ beschreibt diese Sprache.

Der reguläre Ausdruck ab^* beschreibt die Sprache
$\{b, ab, aab, aaab, \ldots\}$.

Der reguläre Ausdruck $(ab)^*$ beschreibt die Sprache
$\{\varepsilon, ab, abab, ababab, \ldots\}$.

Reguläre Ausdrücke

Die folgenden Eigenschaften sind äquivalent:

Die Sprache L ist regulär.

Es existiert ein DEA D, der L akzeptiert.

Es existiert ein NEA N, der L akzeptiert.

Es existiert ein regulärer Ausdruck r, der L beschreibt.

Die verschiedenen Darstellungen von L lassen sich automatisch mittels eines Programms ineinander umrechnen.