Grundlagen der Informatik

- Bisher: Binäre Zahlensysteme und Boolsche Algebra zur Umsetzung von Addition mit logischen Gattern
- In diesem Kapitel:
  - Grundlagen von Grammatiken zur Beschreibung von Programmiersprachen und anderer formaler Sprachen
  - Spezialfall: Reguläre Ausdrücke zum Information Retrieval (Suche)

Grammatiken zur Beschreibung formaler Strukturen

- Grammatik: Systematische Sprachbeschreibung
  - **Syntax**: beschreibt die zulässige Form der Sätze der Sprache
  - **Semantik**: beschreibt die Bedeutung der Sätze
- Eine Grammatik ist eine Menge von Regeln, die bestimmen, welche Sätze zur Sprache gehören und welche nicht.
- In Computersprachen müssen (im Gegensatz zu natürlichen Sprachen) die Regeln streng eingehalten werden.

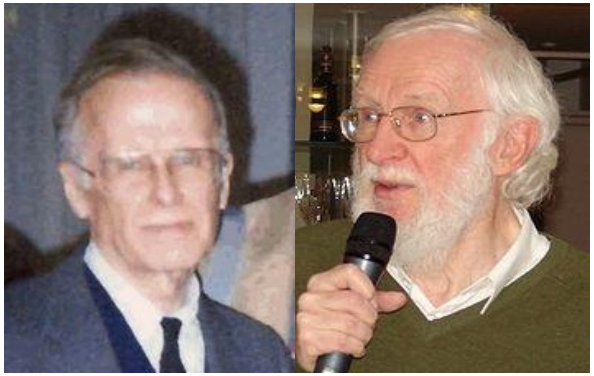
Bestandteile einer Grammatik

- Eine **Grammatik** wird durch 4 Bestandteile definiert:
  - **Terminalsymbole** (T):
    Zeichen der Sprache (was man sieht)
  - **Variablen** (Nicht-Terminalsymbole; V):
    werden durch Aneinanderreihung von T's ersetzt
  - **Regeln** (R): Ersetzungsregeln für alle Variablen
  - **Startsymbol** (S): Eine ausgezeichnete Variable
- **Komplexität der Regeln** bestimmt Ausdrucksstärke und Interpretierbarkeit der Sprachen

Grammatik-Typen

- **Kontextfreie Grammatiken**: Die Bedeutung der Variablen auf der linken Seite einer Regel ist unabhängig vom Kontext, in dem die Variable vorkommt:
  - Auf der linken Seite steht immer nur eine Variable
  - Beispiel: A := B C | D E
  - Gegenbeispiel: 'a' A := 'a' B C und 'b' A := 'b' D E
  - gut interpretierbar
- **Reguläre Grammatiken**:
  Kontextfreie Grammatiken, bei der die rechte Seite einer Regel entweder ein Terminalzeichen oder ein Terminalzeichen gefolgt von einer Variablen enthält :
  - Beispiel: A := 'a' | 'a' A
  - besonders einfach interpretierbar

Beispiel: Extended Backus-Naur-Form



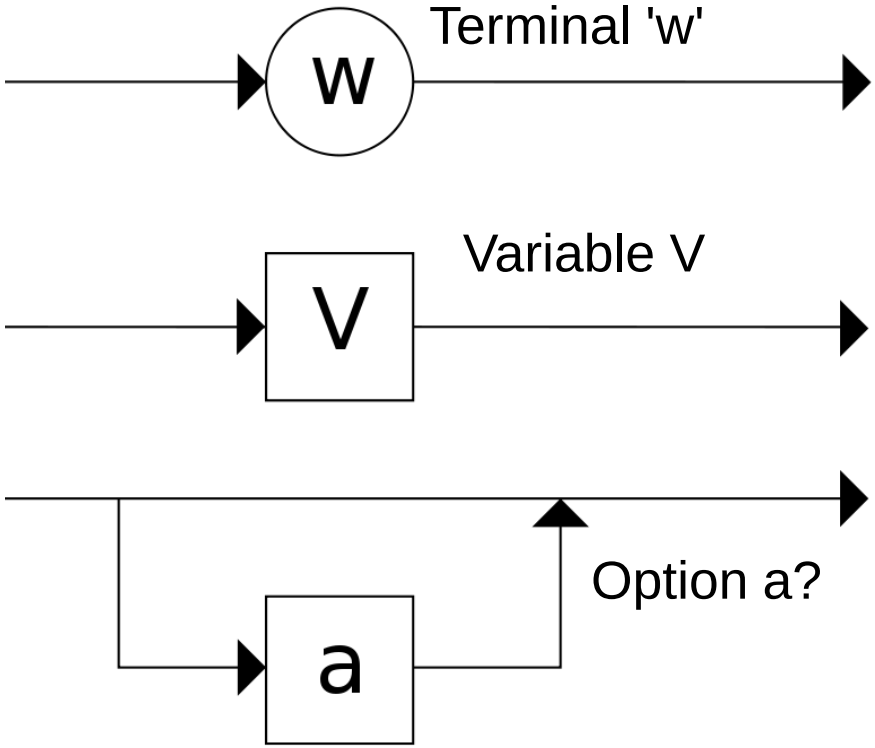
Backus Naur

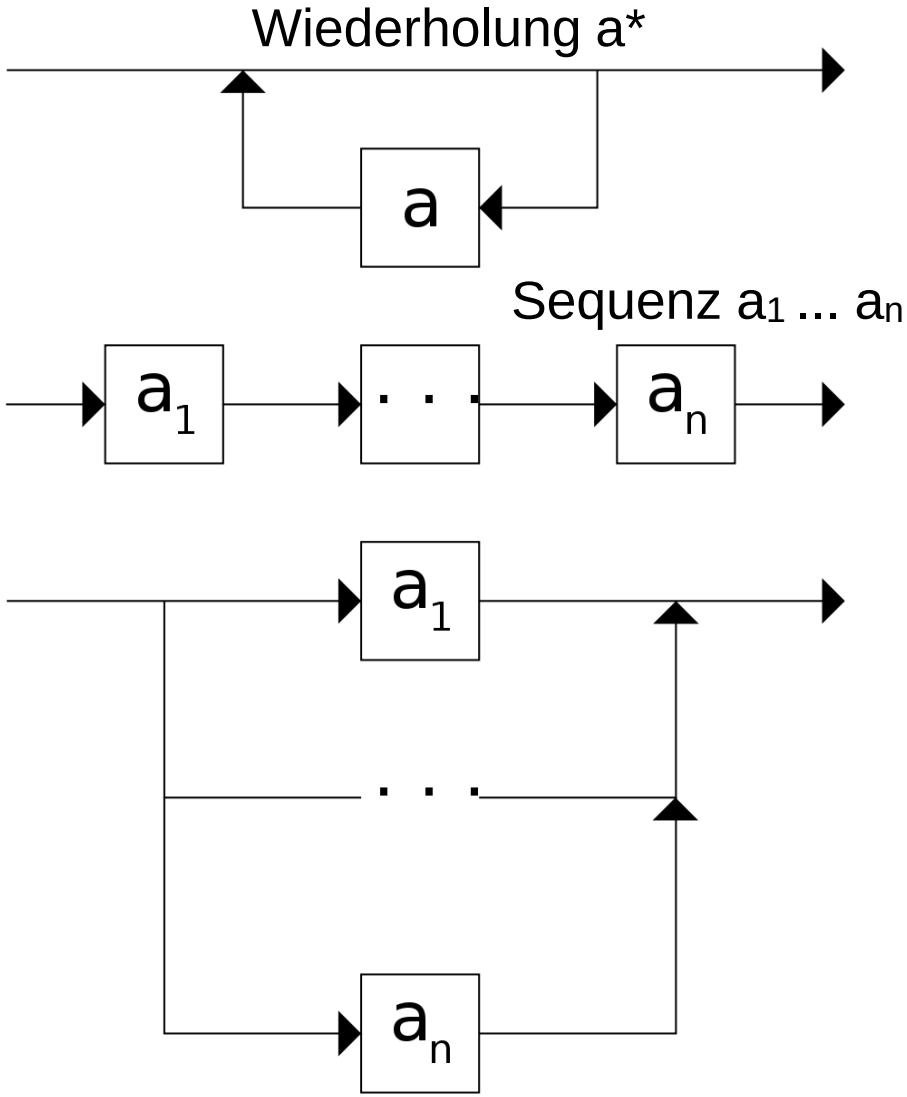
- EBNF: Extended Backus-Naur-Form
- Weit verbreitete Grammatik zur Beschreibung von Sprachen
- 1960 von Backus und Naur zur Beschreibung der Syntax von Programmiersprachen erfunden; später durch u.a. Niklaus Wirth erweitert.

EBNF

- **Kontextfreie Regeln der Art**:
  - "linker Teil" := "Ersetzung durch rechten Teil"
  - z.B. **A := B C D** statt ":=" auch "=" oder "::="
- **Unterscheidung: Variablen & Terminalsymbolen**:
  A := 'b' C 'd' oder <A> := b <C> d bzw. A := "b" C "d"
  - ➢ keine Aufzählung von Variablen und Terminalsymbolen nötig
- **Konstrukte zur Regelbildung**:
  - **Sequenz**: A B C oder mit Kommata A, B, C
  - **Alternative**: A | B | C eine Alternative wird ausgewählt
  - **Option**: A? A darf weggelassen werden (Kardinalität: 0..1); auch: [A]
  - **Wiederholung**: A* A darf weggelassen oder beliebig oft wiederholt werden (Kardinalität 0..N); auch {A}
  - **Mindestens 1-malige Wiederholung**: A+ (Kard.: 1 .. N); auch <A>
  - **Gruppierung**: (A B C) wichtig bei Verschachtelungen wie z.B. (A | B)*

Grafische Darstellung: Syntaxdiagramme





Beispiele für Sprachen

Sprache mit beliebigen Sequenzen von “0” und “1”: $w = \{0, 1\}^*$

- S := ‘0’ | ‘1’ | S’1’ | S’0’ | ε

Palindrome aus “0” und “1”: $w = \{0, 1\}^* \mid w = w^R$ (gespiegelt)

- S := ‘0’S’0’ | ‘1’S’1’ | ‘0’ | ‘1’ | ε

Sprachen mit gleichvielen “0” und “1”: $w = \{0, 1\}^* \mid \quad |0| = |1|$

- S := AB | ε;
- A := ‘0‘S | S‘0‘ | ‘0‘;
- B := ‘1‘S | S‘1‘ | ‘1‘;

(ε = leeres Wort)

Aufgabe1 für Sprachen

ZifferOhneNull := ‘1’|‘2’|‘3’|‘4’|‘5’|‘6’|‘7’|‘8’|‘9’

Ziffer := ZifferOhneNull | ‘0’

Integer := ‘0’ | ‘-’?, ZifferOhneNull, Ziffer*

Buchstabe := ‘A’|B’|…|’Z’|’a’|…|’z’

Wort:= Buchstabe+

Schreiben Sie eine Sprache mit einfachen arithmetischen Ausdrücken, nämlich Addition (+) und Multiplikation (*) mit vollständiger Klammerung, z.B.

*((10+11)*12)*

Lösung für Aufgabe1

ZifferOhneNull := ‘1’|‘2’|‘3’|‘4’|‘5’|‘6’|‘7’|‘8’|‘9’

Ziffer := ZifferOhneNull | ‘0’

Integer := ‘0’ | ‘-’?, ZifferOhneNull, Ziffer*

Buchstabe := ‘A’|B’|…|’Z’|’a’|…|’z’

Wort:= Buchstabe+

Schreiben Sie eine Sprache mit einfachen arithmetischen Ausdrücken, nämlich Addition (+) und Multiplikation (*) mit vollständiger Klammerung, z.B.

*((10+11) * 12)*

S := ‘(‘ S ‘+‘ S ‘)‘ | ‘(‘ S ‘*‘ S ‘)‘ | Integer

oder

Operator := ‘+‘ | ‘*‘
S := ‘(‘ S Operator S ‘)‘ | Integer

Aufgabe2 für Sprachen

Bezeichner := Buchstabe (Buchstabe | Ziffer)*

Satzzeichen := ‘,’ | ‘!’ | ‘.’ | ‘ ‘| …

LZ := ‘ ‘

String: ‘ " ’ (Buchstabe | Ziffer | Satzzeichen)* ‘ " ’

Schreiben Sie eine Grammatik für eine ganz einfache Programmiersprache, die nur Zuweisungen erlaubt, z.B.

```
PROGRAM DEMO1
 BEGIN
   A0:=3;
   B:=45;
   H:=-100023;
   C:=A;
   D123:=B34A;
   ESEL:=GIRAFFE;
   TEXTZEILE:="Hallo, Welt!";
 END.
```

Lösung für Aufgabe2

Bezeichner := Buchstabe (Buchstabe | Ziffer)*

Satzzeichen := ‘,’ | ‘!’ | ‘.’ | ‘ ‘| …

LZ := ‘ ‘

String: ‘ " ’ (Buchstabe | Ziffer | Satzzeichen)* ‘ " ’

Programm := 'PROGRAM' LZ Bezeichner LZ 'BEGIN' LZ (Zuweisung ‘;‘)* 'END.'

Zuweisung := Bezeichner, ‘:=‘, (Zahl | Bezeichner | String) ;

```
PROGRAM DEMO1
 BEGIN
   A0:=3;
   B:=45;
   H:=-100023;
   C:=A;
   D123:=B34A;
   ESEL:=GIRAFFE;
   TEXTZEILE:="Hallo, Welt!";
 END.
```

Beispiel: Grammatik aus Zeitungs-Artikel

Grammatik: (T, V, R, S)

T = { 'A', 'B', 'C', ... , 'Z', 'a', ..., 'z', '1', ..., '9' , '.', ... ,"Pixelgrafik"}

V = { Spiegel-Artikel, Aufreißer, Schlagzeile, Kurzfassung, Inhalt,
Bildteil, Autor, Abschnitt, Bild, Bildunterschrift, Wort, Zeichen}

R = { Spiegel-Artikel := Aufreißer Schlagzeile Autor? Bildteil Kurzfassung Inhalt+
Inhalt := Bildteil | Abschnitt
Bildteil := Bild Bildunterschrift
Aufreißer := Wort+
Schlagzeile := Wort+
Kurzfassung := Wort+
Abschnitt := Wort+
Bildunterschrift := Wort+
Autor := Wort+
Wort := Zeichen+
Zeichen := 'A' | 'B' | 'C' | ... | 'Z' | 'a' | ... | 'z' | '1'| ... | '9' | '.' | '\n'
Bild := ‘Pixelgrafik’ | ‘Video-Plugin’ }

S = Spiegel-Artikel

Schäuble im Glück

Schrottbank entdeckt 55,5 Milliarden Euro

dapd

Finanzminister Schäuble: Klammheimliche Freude

Ein kapitaler Fehler! Finanzminister Schäuble kann sich über unerwarteten Geldsegen freuen: Ein jetzt aufgedeckter Bilanzpfusch bei der Bad Bank der Hypo Real Estate verkleinert das Staatsdefizit. Aus zunächst gemeldeten 24,5 Milliarden Euro an Fehlbuchungen wurden nun sogar 55,5 Milliarden.

München/Berlin - Wolfgang Schäuble ist dafür bekannt, wenig Verständnis für Fehlleistungen zu zeigen - über diesen Pfusch könnte sich der CDU-Finanzminister aber sehr gefreut haben: Weil die Experten bei der Bad Bank der verstaatlichten Hypo Real Estate (HRE) die Verbindlichkeiten in der Bilanz falsch gebucht hatten, ist Deutschland schlagartig um 55,5 Milliarden Euro weniger verschuldet als bislang angenommen.

Beispiel für Ableitung

Aufreißer →

Schlagzeile →

Autor fehlt

Bildteil →

Kurzfassung →

Inhalt →

Schäuble im Glück

Schrottbank entdeckt 55,5 Milliarden Euro

dapd

Finanzminister Schäuble: Klammheimliche Freude

Ein kapitaler Fehler! Finanzminister Schäuble kann sich über unerwarteten Geldsegen freuen: Ein jetzt aufgedeckter Bilanzpfusch bei der Bad Bank der Hypo Real Estate verkleinert das Staatsdefizit. Aus zunächst gemeldeten 24,5 Milliarden Euro an Fehlbuchungen wurden nun sogar 55,5 Milliarden.

München/Berlin - Wolfgang Schäuble ist dafür bekannt, wenig Verständnis für Fehlleistungen zu zeigen - über diesen Pfusch könnte sich der CDU-Finanzminister aber sehr gefreut haben: Weil die Experten bei der Bad Bank der verstaatlichten Hypo Real Estate (HRE) die Verbindlichkeiten in der Bilanz falsch gebucht hatten, ist Deutschland schlagartig um 55,5 Milliarden Euro weniger verschuldet als bislang angenommen.

Beispiel für Ableitung

Spiegel-Artikel := Aufreißer Schlagzeile Autor? Bildteil Kurzfassung Inhalt Inhalt Inhalt Inhalt Inhalt ➞

Wort Wort Wort Schlagzeile Autor? Bild Bildunterschrift Kurzfassung Inhalt Inhalt (...) ➞

Zeichen ... Zeichen Wort Wort Schlagzeile Autor? Bildteil Kurzfassung Inhalt Inhalt (...) ➞

Schäuble Wort Wort Schlagzeile Autor? Bildteil Kurzfassung Inhalt Inhalt (...) ➞

Schäuble im Glück Schlagzeile Autor? Bildteil Kurzfassung Inhalt Inhalt (...) ➞

Schäuble im Glück Wort Wort Wort Wort Wort Autor? Bildteil Kurzfassung Inhalt Inhalt(..).. ➞

Schäuble im Glück Schrottbank entdeckt 55,5 Milliarden Euro Autor? Bildteil Kurzfassung Inhalt Inhalt (...)... ➞

Schäuble im Glück Schrottbank entdeckt 55,5 Milliarden Euro Bildteil Kurzfassung Inhalt Inhalt (...) ➞

Schäuble im Glück Schrottbank entdeckt 55,5 Milliarden Euro Bildteil Kurzfassung Inhalt (...) ➞

Schäuble im Glück Schrottbank entdeckt 55,5 Milliarden Euro. Bild Bildunterschrift Kurzfassung Inhalt... ➞

…

Schäuble im Glück

Schrottbank entdeckt 55,5 Milliarden Euro

dapd

Finanzminister Schäuble: Klammheimliche Freude

Ein kapitaler Fehler! Finanzminister Schäuble kann sich über unerwarteten Geldsegen freuen: Ein jetzt aufgedeckter Bilanzpfusch bei der Bad Bank der Hypo Real Estate verkleinert das Staatsdefizit. Aus zunächst gemeldeten 24,5 Milliarden Euro an Fehlbuchungen wurden nun sogar 55,5 Milliarden.

München/Berlin - Wolfgang Schäuble ist dafür bekannt, wenig Verständnis für Fehlleistungen zu zeigen - über diesen Pfusch könnte sich der CDU-Finanzminister aber sehr gefreut haben: Weil die Experten bei der Bad Bank der verstaatlichten Hypo Real Estate (HRE) die Verbindlichkeiten in der Bilanz falsch gebucht hatten, ist Deutschland schlagartig um 55,5 Milliarden Euro weniger verschuldet als bislang angenommen.

Erweiterungen: Umbruch & Hervorhebung

Spiegel-Artikel := Aufreißer Schlagzeile Autor Bildteil Kurzfassung Inhalt

Inhalt := (Teil Umbruch)* Teil

Teil := Bildteil | Abschnitt

Bildteil := Bild Bildunterschrift

Aufreißer := "Mark-A" Wort+ "End-Mark-A" Umbruch

Schlagzeile := "Mark-S" Wort+ "End-Mark-S" Umbruch

Kurzfassung := "Mark-K" Abschnitt "End-Mark-K" Umbruch

Abschnitt := Satzteil+

Satzteil := Normal | Fetttext | Kursivtext

Normal := Wort+

Fetttext := "Mark-F" Abschnitt "End-Mark-F"

Kursivtext := "Mark-I" Abschnitt "End-Mark-I"

Bildunterschrift := "Mark-B" Abschnitt "End-Mark-B"

Wort := Zeichen+ Zeichen := 'A' | 'B' | 'C' | ... | 'Z' | 'a' | ... | 'z' | '1'| ... | '9' | '.' |

...Bild := "Pixelgrafik“| Video-Plugin

Reguläre Ausdrücke (Regular Expressions = RegEx)

Reguläre Ausdrücke können mit regulären Grammatiken erzeugt werden.

Sie dienen vor allem zur Suche nach bestimmten Mustern in Texten.

Beispiele:

M[ae][iy]er -> *Maier*, *Meier*, *Mayer*, *Meyer*

.*@uni-wuerzburg.de -> alle Email-Adressen, die auf *@uni-wuerzburg.de* enden

[12][0-9][0-9] -> Alle Jahreszahlen zwischen 1000 und 2999

1[0-9]{3}|20[01][0-8] -> Alle Jahreszahlen zwischen 1000 und 2018

[0-3][0-9]\.[01][0-9]\.[12][0-9]{3} -> Alle Datumsangaben (und mehr), z.B. *05.05.1955*

Hunde?s? -> *Hund*, *Hunde*, *Hundes* und auch *Hunds*

Hand.* -> *Hand*, *Handschuh*, usw. aber auch *Hand und Fuss …*

Hand.*\w+ -> *Hand*, *Handschuh*, nur Wörter die mit „Hand“ beginnen

Formale Definition regulärer Ausdrücke

Reguläre Ausdrücke sind immer über einem vorgegebenen Zeichenvorrat Σ definiert, dem sogenannten Alphabet (entspricht Terminalsymbole = T). Reguläre Ausdrücke basieren auf genau drei Operationen: Alternative, Verkettung und Wiederholung. Die formale Definition sieht folgendermaßen aus:

1. ∅ (das spezielle Symbol für die leere Menge) ist ein regulärer Ausdruck.
2. für alle $a_i \in \Sigma$ ist a_i (die Repräsentation eines Zeichens aus dem zugrunde liegenden Alphabet) ein regulärer Ausdruck.
3. Sind x und y reguläre Ausdrücke, so sind auch (x|y) (Alternative), (xy) (Verkettung) und (x^*) (Wiederholung, beliebig oft, auch 0 Mal) reguläre Ausdrücke.

Reguläre Ausdrücke in der Praxis

Reguläre Ausdrücke wurden früher in Suchmaschinen wie Google und Suchfunktionen standardmäßig angeboten, derzeit aber nur teilweise, weil sie für viele Nutzer zu kompliziert sind.

Im folgenden geben wir eine Übersicht über wichtige Konstrukte (Darstellung aus Wikipedia)

RegEx-Symbole: Zeichenauswahl

`[egh]`	eines der Zeichen „e“, „g“ oder „h“
`[0-6]`	eine Ziffer von „0“ bis „6“ (Bindestriche sind Indikator für einen Bereich)
`[A-Za-z0-9]`	ein beliebiger lateinischer Buchstabe oder eine beliebige Ziffer
`[^a]`	ein beliebiges Zeichen außer „a“ („^“ am Anfang einer Zeichenklasse negiert selbige)
`[-A-Z]`, `[A-Z-]` (bzw. `[A-Z\-a-z]`, allerdings nicht gemäß POSIX)[5]	Auswahl enthält auch den Bindestrich „-“, wenn er das erste oder das letzte Zeichen in der Aufzählung einer Zeichenklasse ist bzw. bei PCRE, wenn seine Metafunktion innerhalb einer Auswahl durch ein vorangestelltes „\“-Zeichen aufgehoben wird

RegEx-Symbole: Vordefinierte Zeichenklassen

`\d`	**d**igit	eine Ziffer, also [0-9] (und evtl. auch weitere Zahlzeichen in Unicode, z. B. bengalische Ziffern)
`\D`	no **d**igit	ein Zeichen, das keine Ziffer ist, also [^\d]
`\w`	**w**ord character	ein Buchstabe, eine Ziffer oder der Unterstrich, also [a-zA-Z_0-9] (und evtl. auch nicht-lateinische Buchstaben, z. B. Umlaute)
`\W`	no **w**ord character	ein Zeichen, das weder Buchstabe noch Zahl noch Unterstrich ist, also [^\w]
`\s`	white**s**pace	meist zumindest das Leerzeichen und die Klasse der Steuerzeichen \f, \n, \r, \t und \v
`\S`	no white**s**pace	ein Zeichen, das kein Whitespace ist, also [^\s]

RegEx-Symbole: Quantoren

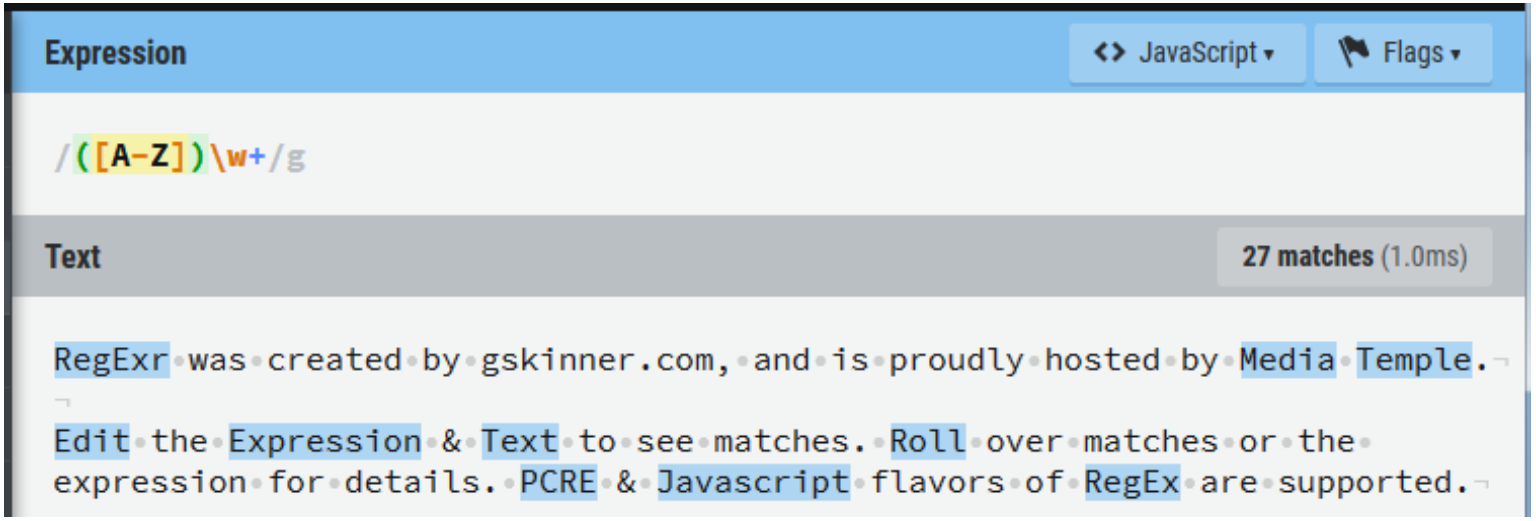
`?`	Der voranstehende Ausdruck ist optional, er kann einmal vorkommen, braucht es aber nicht, das heißt, der Ausdruck kommt null- oder einmal vor. (Dies entspricht `{0,1}`)
`+`	Der voranstehende Ausdruck muss mindestens einmal vorkommen, darf aber auch mehrfach vorkommen. (Dies entspricht `{1,}`)
`*`	Der voranstehende Ausdruck darf beliebig oft (auch keinmal) vorkommen. (Dies entspricht `{0,}`)
`{n}`	Der voranstehende Ausdruck muss exakt *n*-mal vorkommen. (Dies entspricht `{n,n}`)
`{min,}`	Der voranstehende Ausdruck muss mindestens *min*-mal vorkommen.
`{min,max}`	Der voranstehende Ausdruck muss mindestens *min*-mal und darf maximal *max*-mal vorkommen.
`{0,max}`	Der voranstehende Ausdruck darf maximal *max*-mal vorkommen.

RegEx-Symbole: Weitere Zeichen

`^`	steht für den Zeilenanfang (nicht zu verwechseln mit `^` bei der Zeichenauswahl mittels `[` und `]`).
`$`	kann je nach Kontext für das Zeilen- oder Zeichenketten-Ende stehen, wobei bei manchen Implementierungen noch ein „\n“ folgen darf. Das tatsächliche Ende passt zu `\z` .
`\`	hebt gegebenenfalls die Metabedeutung des nächsten Zeichens auf (siehe Maskierungszeichen). Beispielsweise lässt der Ausdruck `(A\*)+` die Zeichenketten „A*“, „A*A*“ usw. zu. Auf diese Weise lässt sich auch ein Punkt „.“ mit `\.` suchen, während nach `\` mit `\\` gesucht wird.
`\b`	leere Zeichenkette am Wortanfang oder am Wortende
`\B`	leere Zeichenkette, die *nicht* den Anfang oder das Ende eines Wortes bildet
`\<`	leere Zeichenkette am Wortanfang
`\>`	leere Zeichenkette am Wortende
`\n`	ein Zeilenumbruch im Unix-Format
`\r`	ein Zeilenumbruch im (alten, d. h. vor dem Jahr 1999) Mac-Format
`\r\n`	ein Zeilenumbruch im DOS- und Windows-Format
`\t`	ein Horizontal-Tabulatorzeichen

Tools zum Üben mit Regulären Ausdrücken

https://regexr.com/



https://www.python-kurs.eu/re.php

```
>>> import re
>>> line = "He is a German called Mayer."
>>> if re.search(r"M[ae][iy]er",line): print "I found one!"
...
I found one!
>>>
```

Beispiele für reguläre Ausdrücke

Mietkosten: (\d{1,4}(\.|,)-)
Wohnungsfläche:
Anzahl Zimmer:
Telefonnummer:
Ort:

Margetshöchheim, sonnige 4-ZW, 96 m2, 1. OG, Bad neu, Balk., Keller, Garten, Garage, Energieausweis 114 KwH, KM 720.- EUR + NK, Hobbyraum 20 m2 im Keller zusätzl. mögl., ab 01.01.2017. Tel. 0157/32241087
Zellingen-Retzbach, 4 ZW, 1.OG, 105 m2, energetisch saniert, renoviert, bezugsfertig, Balkon + Stellplatz, 600.- + 120.- NK. Tel. 0179/6616138
Arnstein - komf. 4-Zi-Whg, 95m2, Parkett, EBK, sep. WC, Energiepass, Balkon, Garage, Keller, ab 01.01.17. Tel. 09353/99284 ab 10.10.2016
TG-Stellplatz in Lindenbrunnenweg/Heinrich-Beck-Str. ab sof. zvm.,T. 09721/70450
4-ZW, Kürnach, 93 m2, 1. OG, Balk., Keller, 2 Stellpl., EnBW 77,1 kwH. 0172/6612830
Möbl. 2-Zi.-Whg., ca. 63 m2, sep. Eingang, SW-Deutschhof, Geschäfte i.d. Nähe, an alleinsteh. deutschspr. Person (NR), keine Haustiere, WM 450,- EUR, Tel. 09721/803206
WÜ/City, Rottendorfer Str., sehr hübsche 2 1/2-ZW, 74 m2, Altbau, offener Wohnbereich, Parkett, mod., hochw. EBK, Keller, 710.- EUR + 100.- EUR NK, zum 1.12., Besichtigung am 22.10.2016. Tel. 0160/95420408
4-ZW, WÜ-Sanderau, Blk., ZH, 88 m2, Garage, 905.- + NK. Tel. 0163/7183067
Hettstadt 2 ZW, 57 m2, Blk, Abstellr., Keller, Stellpl, 350.- + NK ab 1.11.16 od. später. EBK kann übern. werden 0931/4675499
Ochsenfurt, Tückelhäuser Strasse, 40 m², Büro-, Praxis-,Gewerberäume, sep. Eingang, Stellplatz, EUR/m² 5,00 + NK zu vermieten. Tel. 0176/51169145