Mustervergleich mit Regulären Ausdrücken

- ▷ Platzhalter
- ▷ Mustervergleich
- ▷ Reguläre Ausdrücke
- ▷ Typische Anwendungen
- ▷ Ungewöhnliche Anwendungen
- ▷ Endliche Automaten
- ▷ Endliche Automaten mit Ausgabe
- ▷ Nichtdeterministische endliche Automaten
- ▷ Erweiterungen

Platzhalter

UNIX-Filesystem: Hierarchische Datenbank aller Daten und Programme

Namenskonventionen: Ein Dateiname ist eine beliebige Zeichenkette, die keines der folgenden Sonderzeichen enthält:

? * $ { }

/ ; \ ( )

' " ` < >

& { }

- ☐ beliebig lang
- ☐ Punkt hat keinen Sonderstatus

  Ausnahmen: Punkt am Anfang
  - o relative Verzeichnisangabe
  - o versteckte Files

Platzhalter (Wildcards): Verkürzende Notation für beliebige Zeichen oder Zeichengruppen (Maskierung)

?	steht für ein beliebiges Zeichen
*	steht für eine beliebige Zeichenfolge (KLEENE-Star)
[abc]	steht für eines der Zeichen **a**, **b** oder **c**
[a-c]	steht für eines der Zeichen aus dem ascii-Bereich von **a** bis **c**

Mustervergleich

Mustervergleich (Pattern Matching): Vergleich eines Musters (Zeichenkette mit Platzhaltern) mit einer Zielsequenz (reine Zeichenkette)

$$\begin{array}{ccc} \text{Pattern} & \rightarrow & \text{Target} \\ \text{Muster} & \rightarrow & \text{Zielsequenz} \end{array}$$

```
nats18 <31> ls
21fragen      Konzept.kap4  kap2.tex      kap4.tex
Konzept.kap1  kap1.ps.gz    kap3.ps.gz    lit.bib
Konzept.kap2  kap1.tex      kap3.tex      vorw.ps.gz
Konzept.kap3  kap2.ps.gz    kap4.ps.gz    vorw.tex
nats18 <32> ls *1*
21fragen      Konzept.kap1  kap1.ps.gz    kap1.tex
nats18 <33> ls *kap1*
Konzept.kap1  kap1.ps.gz    kap1.tex
nats18 <34> ls *k??1*
Konzept.kap1  kap1.ps.gz    kap1.tex
nats18 <35> ls *??1*
Konzept.kap1  kap1.ps.gz    kap1.tex
nats18 <36> ls *k??[23].*
kap2.ps.gz  kap2.tex    kap3.ps.gz  kap3.tex
nats18 <37> ls *k??[2-4].*
kap2.ps.gz  kap3.ps.gz  kap4.ps.gz
kap2.tex    kap3.tex    kap4.tex
```

totaler Mustervergleich: Muster muß die Zielsequenz vollständig beschreiben.

```
nats18 <38> nats18 <37> ls *.ps
No match
nats18 <39> ls *.ps*
kap1.ps.gz  kap3.ps.gz  vorw.ps.gz
kap2.ps.gz  kap4.ps.gz
```

Prozeduraufruf: $h(m, s)$

- ☐ Vergleicht Muster m mit Zielsequenz s
- ☐ Liefert als Resultat eine Erfolgsmeldung

$$h(m, s) = \begin{cases} \text{TRUE} & \text{falls Muster paßt} \\ \text{FALSE} & \text{falls Muster nicht paßt} \end{cases}$$

partieller Mustervergleich: Muster beschreibt eine nichtleere Teilsequenz
Start des Mustervergleichs an jeder Position der Zielsequenz

Prozeduraufruf: $g(m, s)$

$$g(m, s) = \begin{cases} \text{TRUE} & \text{falls} \quad \exists s' = \text{substring}(s) \; . \; h(m, s') \\ \text{FALSE} & \text{sonst} \end{cases}$$

Implementierung: `grep <muster> <file>`

- □ `<file>` enthält beliebig viele Zielsequenzen (Zeilen)

```
nats18 <10> cat test1
abc
bcd
cde
def
efg
fgh
ghi
nats18 <11> cat test2
ac
abc
axb
axxb
axxxb
axxxxb
```

- □ Resultat: Alle Zeilen (ggf. keine), in denen das Muster an einer beliebigen Position paßt.

Reguläre Ausdrücke

Definition eines Formalismus (Sprache): Formale Spezifikation von Form und Bedeutung der Sprachkonstrukte.

Form	Repräsentation	Syntax
$\updownarrow$	$\updownarrow$	$\updownarrow$
Bedeutung	Interpretation	Semantik

Zielsequenz: Worte über einem Alphabet Z

$s = z_1 \ldots z_n \qquad z_i \in Z, i = 1, \ldots, n$

$s \in Z^*$

Muster: Regulärer Ausdruck (RA)
rekursive Definition von Syntax und Semantik

Definition: Reguläre Ausdrücke (1)

Einzeichenausdruck (EA):

Syntax	Ein Einzeichenausdruck r ist ein RA.
Semantik	—

Zeichen:

Syntax	Ein Zeichen $z \in Z$ ist ein EA.
Semantik	$h(z, s) = \begin{cases} \text{TRUE} & \text{falls} \quad s = z \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die das Zeichen `d` enthalten.

	`nats18 <12> grep 'd' test1`
	`bcd`
	`cde`
	`def`

Definition: Reguläre Ausdrücke (2)

Universelles Zeichen:

Syntax	Das Zeichen . (außerhalb eines Auswahlausdrucks) ist ein EA.
Semantik	$h(., s) = \begin{cases} \text{TRUE} & \text{für beliebiges einstelliges } s \\ \text{FALSE} & \text{sonst} \end{cases}$

Zeichenauswahl:

Syntax	Eine Zeichenauswahl $[z_1, \ldots, z_n], z_i \in Z$ ist ein EA.
Semantik	$h([z_1, \ldots z_n], s) = \begin{cases} \text{TRUE} & \text{falls} \quad \exists i.s = z_i \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die ein Zeichen enthalten, das einem Vokal entspricht.

	`nats18 <13> grep '[aeiou]' test1`
	`abc`
	`cde`
	`def`
	`efg`
	`ghi`

Definition: Reguläre Ausdrücke (3)

Negative Zeichenauswahl:

Syntax	Eine negative Zeichenauswahl $[\^{}z_1, \ldots, z_n], z_i \in Z$ ist ein EA.
Semantik	$h([\^{}z_1, \ldots z_n], s) = \begin{cases} \text{TRUE} & \text{falls} \quad \neg\exists i . s = z_i \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die ein Zeichen enthalten, das nicht einem Konsonanten entspricht.
	nats18 <14> grep '[^bcdfgh]' test1 abc cde def efg ghi
	!!! Nicht: Finde alle Zeilen, die keinen Konsonanten enthalten!

Definition: Reguläre Ausdrücke (4)

Bereichsauswahl:

Syntax	Eine Bereichsauswahl $[z_u - z_o], z_u, z_o \in Z$, ascii($z_u$) $<$ ascii(z_o) ist ein EA
Semantik	$h([z_u - z_o], s) = \begin{cases} \text{TRUE} & \text{falls} \quad \exists i.s = z_i \\ & \wedge u \leq i \leq o \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die ein Zeichen zwischen a und c (einschließlich) enthalten.

	`nats18 <15> grep '[a-c]' test1`
	`abc`
	`bcd`
	`cde`

Definition: Reguläre Ausdrücke (5)

Klammerung (GRA):

Syntax	Ein geklammerter RA $\backslash(r\backslash)$ ist ein RA (`egrep` auch vereinfacht zu (r))
Semantik	$h(\backslash(r\backslash), s) = \begin{cases} \text{TRUE} & \text{falls} \quad h(r, s) \\ \text{FALSE} & \text{sonst} \end{cases}$

Beispiel	`nats18 <15> grep '\([a-c]\)' test1`
	`abc`
	`bcd`
	`cde`

Definition: Reguläre Ausdrücke (6)

Verkettung: (Produkt)

Syntax	Eine Verkettung von EAs oder GRAs r_1r_2 ist ein RA
Semantik	$h(r_1r_2, s) = \begin{cases} \text{TRUE} \quad \text{falls} & s = s_1s_2 \wedge s_i \in Z^* \\ & \wedge\, h(r_1, s_1) \wedge h(r_2, s_2) \\ \text{FALSE} \quad \text{sonst} & \end{cases}$
Beispiel	Finde alle Zeilen, die eine bestimmte Menge von Zeichenpaaren enthalten: $\{$bc$\}$, $\{$ac, ad, ae, ...$\}$, ...

	`nats18 <134> grep 'bc' test1`
	`abc`
	`bcd`
	`nats18 <135> grep '[abc][cde]' test1`
	`abc`
	`bcd`
	`cde`
	`nats18 <136> grep '[abc][^abc]' test1`
	`bcd`
	`cde`

Definition: Reguläre Ausdrücke (7)

Wiederholung: (Iteration)

Syntax	Eine beliebig häufige (auch 0-fache) Wiederholung eines EAs (oder GRAs) $r*$ ist ein RA
Semantik	$h(r*, s) = \begin{cases} \text{TRUE} & \text{falls} \quad s = s_1 s_2 \ldots s_n \\ & \quad \wedge \forall i = 0, \ldots, n. \\ & \quad\quad (s_i \in Z^* \wedge h(r, s_i)) \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die zwischen den Zeichen `a` und `b` kein, ein oder mehrere Vorkommen der Zeichenkette `x` bzw. beliebiger Zeichen außer `x` enthalten.

	`nats18 <161> grep 'ax*b' test2`
	`abc`
	`axb`
	`axxb`
	`axxxb`
	`axxxxb`
	`nats18 <163> grep 'a[^x]*b' test2`
	`abc`

Variante (nur `egrep`): + verhält sich wie *, aber mindestens einmaliges Vorkommen

Definition: Reguläre Ausdrücke (8)

Eingeschränkte Wiederholung: (nur `grep`)

Syntax	Eine eingeschränkt häufige Wiederholung eines EAs (oder GRAs) $r\backslash\{u,o\backslash\}$ ist ein RA
Semantik	$h(r\backslash\{u,o\backslash\},s) = \begin{cases} \text{TRUE} & \text{falls} \quad s = s_1 s_2 \ldots s_n \\ & \wedge\ u \leq n \leq o \\ & \wedge \forall i = 1,\ldots,n. \\ & \quad (s_i \in Z^* \\ & \quad\quad \wedge\ h(r,s_i)) \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die zwischen den Zeichen `a` und `b` zwei bis drei, genau zwei, bzw. mindestens zwei Vorkommen der Zeichenkette `x` enthalten.

	`nats18 <164> grep 'ax\{2,3\}b' test2`
	`axxb`
	`axxxb`
	`nats18 <167> grep 'ax\{2\}b' test2`
	`axxb`
	`nats18 <168> grep 'ax\{2,\}b' test2`
	`axxb`
	`axxxb`
	`axxxxb`

Definition: Reguläre Ausdrücke (9)

Eingeschränkte Wiederholung: (Fortsetzung)

Beispiel	Finde alle Zeilen, die zwei Vorkommen der Zeichenkette `x` enthalten.

	`nats18 <169> grep 'x\{2\}' test2`
	`axxb`
	`axxxb`
	`axxxxb`

Alternative (nur `egrep`): (Vereinigung)

Syntax	Eine Alternative zweier RAs $r_1 \vert r_2$ ist ein RA
Semantik	$h(r_1 \vert r_2, s) = \begin{cases} \text{TRUE} & \text{falls} \quad h(r_1, s) \vee h(r_2, s) \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	Finde alle Zeilen, die die Zeichenfolge `abc` oder `def` enthalten.

	`nats18 <170> egrep 'abc|def' test1`
	`abc`
	`def`

Definition: Reguläre Ausdrücke (10)

eingeschränkt partieller Vergleich: Verankerung

Syntax	Ein links verankerter RA $\hat{}r$ ist ein RA ^ muß erstes Zeichen des RA sein
Semantik	$g(\hat{}r, s) = \begin{cases} \text{TRUE} & \text{falls} \quad \exists s' = \text{prefix}(s) \;.\; h(r, s') \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	`nats18 <46> grep '^d' test1` `def`
Syntax	Ein rechts verankerter RA $r\$$ ist ein RA $ muß letztes Zeichen des RA sein
Semantik	$g(r\$, s) = \begin{cases} \text{TRUE} & \text{falls} \quad \exists s' = \text{suffix}(s) \;.\; h(r, s') \\ \text{FALSE} & \text{sonst} \end{cases}$
Beispiel	`nats18 <47> grep 'd$' test1` `bcd`

Typische Anwendungen

- Recherche in ASCII-Texten
- Suche in Korpora für natürliche Sprache
- Korpuslinguistische Werkzeuge
- Spezifikation neuer Wörter für die Handschrifterkennung (Newton)
- UNIX-Werkzeuge
  - Suchen: `grep`, `egrep`, `fgrep`
  - Suchen und ersetzen: `sed`
  - Komplexe Texttransformationsaufgaben: `awk`, `perl`

Textrecherche

```
nats18 <74> cat text
 Exzentriker sind ges"under, weil sie gl"ucklicher sind, glaubt
 Weeks in Hunderten von Interviews festgestellt zu haben. Die K"auze
 gingen beispielsweise nur alle acht Jahre zum Arzt, gew"ohnliche
 Naturen m"u"sten sich sechzehnmal h"aufiger in medizinische Behandlung
 begeben. Da die fr"ohlichen Verr"uckten keine Konventionen kennen und
 sich nicht um das Geschw"atz anderer Menschen k"ummern, haben sie
 weniger Stre"s und mehr Lebensfreude als ihre konformen Mitmenschen,
 erkl"art der Psychologe die Vitalit"at der Sonderlinge.

 Doch nicht nur mit Gesundheit, auch mit Intelligenz scheinen sie
 gesegnet zu sein. Britische Exzentriker haben im Durchschnitt einen
 Intelligenzquotienten von 115 bis 120, hat Weeks mit seinen Tests
 ermittelt. Damit z"ahlten die Typen zu den begabtesten zehn bis
 f"unfzehn Prozent der Bev"olkerung.

nats18 <77> grep 'und[ ,.]' text
 begeben. Da die fr"ohlichen Verr"uckten keine Konventionen kennen und
 weniger Stre"s und mehr Lebensfreude als ihre konformen Mitmenschen,
nats18 <80> egrep ' d(e[mnrs]|ie|as)[ ,.]' text
 begeben. Da die fr"ohlichen Verr"uckten keine Konventionen kennen und
 sich nicht um das Geschw"atz anderer Menschen k"ummern, haben sie
 erkl"art der Psychologe die Vitalit"at der Sonderlinge.
 ermittelt. Damit z"ahlten die Typen zu den begabtesten zehn bis
 f"unfzehn Prozent der Bev"olkerung.
```

Wieviele bestimmte Artikel sind in einem Text enthalten?

1. Beseitigen der Interpunktion

```
nats18 <133> sed 's/[,.]//g' text
 Exzentriker sind ges"under weil sie gl"ucklicher sind glaubt
 Weeks in Hunderten von Interviews festgestellt zu haben Die K"auze
 gingen beispielsweise nur alle acht Jahre zum Arzt gew"ohnliche
 Naturen m"u"sten sich sechzehnmal h"aufiger in medizinische Behandlung
 begeben Da die fr"ohlichen Verr"uckten keine Konventionen kennen und
 sich nicht um das Geschw"atz anderer Menschen k"ummern haben sie
 weniger Stre"s und mehr Lebensfreude als ihre konformen Mitmenschen
 erkl"art der Psychologe die Vitalit"at der Sonderlinge

 Doch nicht nur mit Gesundheit auch mit Intelligenz scheinen sie
 gesegnet zu sein Britische Exzentriker haben im Durchschnitt einen
 Intelligenzquotienten von 115 bis 120 hat Weeks mit seinen Tests
 ermittelt Damit z"ahlten die Typen zu den begabtesten zehn bis
 f"unfzehn Prozent der Bev"olkerung
```

2. Verdoppeln der Wortzwischenräume.

```
nats18 <133> sed 's/[,.]//g;s/ /  /g' text
  Exzentriker  sind  ges"under  weil  sie  gl"ucklicher  sind  glaubt
  Weeks  in  Hunderten  von  Interviews  festgestellt  zu  haben  Die  K"
  gingen  beispielsweise  nur  alle  acht  Jahre  zum  Arzt  gew"ohnliche
  Naturen  m"u"sten  sich  sechzehnmal  h"aufiger  in  medizinische  Beha
  begeben  Da  die  fr"ohlichen  Verr"uckten  keine  Konventionen  kennen
  sich  nicht  um  das  Geschw"atz  anderer  Menschen  k"ummern  haben  s
  weniger  Stre"s  und  mehr  Lebensfreude  als  ihre  konformen  Mitmens
  erkl"art  der  Psychologe  die  Vitalit"at  der  Sonderlinge

  Doch  nicht  nur  mit  Gesundheit  auch  mit  Intelligenz  scheinen  si
  gesegnet  zu  sein  Britische  Exzentriker  haben  im  Durchschnitt  ei
  Intelligenzquotienten  von  115  bis  120  hat  Weeks  mit  seinen  Tes
  ermittelt  Damit  z"ahlten  die  Typen  zu  den  begabtesten  zehn  bis
  f"unfzehn  Prozent  der  Bev"olkerung
```

3. Markieren der bestimmten Artikel.

```
nats18 <133> sed 's/[,.]//g;s/ /  /g;s/ [dD][aei][rnmse] / <&> /g' text
  Exzentriker  sind  ges"under  weil  sie  gl"ucklicher  sind  glaubt
  Weeks  in  Hunderten  von  Interviews  festgestellt  zu  haben  < Die >
  gingen  beispielsweise  nur  alle  acht  Jahre  zum  Arzt  gew"ohnliche
  Naturen  m"u"sten  sich  sechzehnmal  h"aufiger  in  medizinische  Beha
  begeben  Da  < die >  fr"ohlichen  Verr"uckten  keine  Konventionen  ke
  sich  nicht  um  < das >  Geschw"atz  anderer  Menschen  k"ummern  habe
  weniger  Stre"s  und  mehr  Lebensfreude  als  ihre  konformen  Mitmens
  erkl"art  < der >  Psychologe  < die >  Vitalit"at  < der >  Sonderling

  Doch  nicht  nur  mit  Gesundheit  auch  mit  Intelligenz  scheinen  si
  gesegnet  zu  sein  Britische  Exzentriker  haben  im  Durchschnitt  ei
  Intelligenzquotienten  von  115  bis  120  hat  Weeks  mit  seinen  Tes
  ermittelt  Damit  z"ahlten  < die >  Typen  zu  < den >  begabtesten  z
  f"unfzehn  Prozent  < der >  Bev"olkerung
```

4. Markieren von Zeilenanfang und -ende

```
nats18 <133> sed 's/[,.]//g;s/ /  /g;s/ [dD][aei][rnmse] / <&> /g;
s/^ /> /g;s/ $/ </g' text
>  Exzentriker  sind  ges"under  weil  sie  gl"ucklicher  sind  glaubt  <
>  Weeks  in  Hunderten  von  Interviews  festgestellt  zu  haben  < Die
>  gingen  beispielsweise  nur  alle  acht  Jahre  zum  Arzt  gew"ohnlich
>  Naturen  m"u"sten  sich  sechzehnmal  h"aufiger  in  medizinische  Beh
>  begeben  Da  < die >  fr"ohlichen  Verr"uckten  keine  Konventionen  k
>  sich  nicht  um  < das >  Geschw"atz  anderer  Menschen  k"ummern  hab
>  weniger  Stre"s  und  mehr  Lebensfreude  als  ihre  konformen  Mitmen
>  erkl"art  < der >  Psychologe  < die >  Vitalit"at  < der >  Sonderlin

>  Doch  nicht  nur  mit  Gesundheit  auch  mit  Intelligenz  scheinen  s
>  gesegnet  zu  sein  Britische  Exzentriker  haben  im  Durchschnitt  e
>  Intelligenzquotienten  von  115  bis  120  hat  Weeks  mit  seinen  Te
>  ermittelt  Damit  z"ahlten  < die >  Typen  zu  < den >  begabtesten
>  f"unfzehn  Prozent  < der >  Bev"olkerung  <
```

5. Löschen der nicht markierten Textteile.

```
nats18 <133> sed 's/[,.]//g;s/ /  /g;s/ [dD][aei][rnmse] / <&> /g;
s/^ /> /g;s/ $/ </g;s/>[^<]*<//g' text

 Die


 die
 das

 der  die  der




 die  den
 der
```

6. Zählen der Wortvorkommen.

```
nats18 <133> sed 's/[,.]//g;s/ /  /g;s/ [dD][aei][rnmse] / <&> /g;
s/^ /> /g;s/ $/ </g;s/>[^<]*<//g' text | wc
      15       9      60
```

7. Zusammenfassen zu einem sed-Skript

```
nats18 <133> cat b-art
s/[.,]//g
s/ /  /g
s/ [dD][aei][rnmse] / <&> /g
s/^ /> /g
s/ $/ </g
s/>[^<]*<//g
nats18 <133> sed -f b-art text | wc
      15       9      60
```

Korpusrecherche

Korpus: Für linguistische Untersuchungen aufbereitete Textsammlung

Standardkonvention: SGML (Standard Generalised Markup Language)

Beispiel: Syntaktisches Wortarten-Tagging

```
nats18 <140> cat text-tag
 Exzentriker<N> sind<VAUX> ges"under<ADV>, weil<KONJ> sie<PPRO>
 gl"ucklicher<ADJ> sind<VAUX>, glaubt<VFIN> Weeks<PN> in<P>
 Hunderten<N> von<P> Interviews<P> festgestellt<VPPART> zu<PART>
 haben<VAUX>. Die<ART> K"auze<N> gingen<VFIN> beispielsweise<ADV>
 nur<ADV> alle<DET> acht<NUM> Jahre<N> zum<P> Arzt<N>, gew"ohnliche<ADJ>
 Naturen<N> m"u"sten<VAUX> sich<REFL> sechzehnmal<ADV> h"aufiger<ADV>
 in<P> medizinische<ADJ> Behandlung<N> begeben<VINF>. Da<KONJ> die<ART>
 fr"ohlichen<ADJ> Verr"uckten<N> keine<ART> Konventionen<N> kennen<VFIN>
 und<KONJ> sich<REFL> nicht<NEG> um<P> das<ART> Geschw"atz<N>
 anderer<ADJ> Menschen<N> k"ummern<VFIN>, haben<VFIN> sie<PPRO>
 weniger<ADJ> Stre"s<N> und<KONJ> mehr<ADV> Lebensfreude<N> als<KONJ>
 ihre<PPRO> konformen<ADJ> Mitmenschen<N>, erkl"art<VFIN> der<ART>
 Psychologe<N> die<ART> Vitalit"at<N> der<ART> Sonderlinge<N>.

 Doch<KONJ> nicht<NEG> nur<ADV> mit<P> Gesundheit<N>, auch<ADV> mit<P>
 Intelligenz<N> scheinen<VFIN> sie<PPRO> gesegnet<VPPART> zu<Part>
 sein<VAUX>. Britische<ADJ> Exzentriker<N> haben<VAUX> im<P>
 Durchschnitt<N> einen<ART> Intelligenzquotienten<N> von<P> 115<NUM>
 bis<P> 120<NUM>, hat<VAUX> Weeks<PN> mit<P> seinen<POSSP> Tests<N>
 ermittelt<VPPART>. Damit<KONJ> z"ahlten<VFIN> die<ART> Typen<N> zu<P>
 den<ART> begabtesten<ART> zehn<NUM> bis<P> f"unfzehn<NUM> Prozent<N>
 der<ART> Bev"olkerung<N>.
```

Beipiel: Suche aller Verbformen

```
nats18 <6> sed 's/ [^<]*<[^V][^>]*>//g' text-tag
 sind<VAUX>,
 sind<VAUX>, glaubt<VFIN>
 festgestellt<VPPART>
 haben<VAUX>. gingen<VFIN>
,
 m"u"sten<VAUX>
 begeben<VINF>.
 kennen<VFIN>

 k"ummern<VFIN>, haben<VFIN>

, erkl"art<VFIN>
.


,
 scheinen<VFIN> gesegnet<VPPART>
 sein<VAUX>. haben<VAUX>

, hat<VAUX>
 ermittelt<VPPART>. z"ahlten<VFIN>


.
```

Untypische Anwendungen

ASCII-Datenbanken: Strukturierte Datensammlungen als sequentielle ASCII-Datei

Beispiel: Personaldaten

```
nats18 <22> cat personal
Name    Vorname Gebjahr Abteil. Gehalt
Maier   Klaus   35      A3      7500
Schulze Frieda  53      B2      8100
Heine   Hans    61      C1      4300
Gross   Maria   44      C2      5500
Moll    Grit    71      D3      3500
```

Anfrage: Welche Mitarbeiterin ist 1935 geboren?

```
nats18 <180> grep 35 personal
Maier   Klaus   35      A3      7500
Moll    Grit    37      D3      3500
```

→ Anfrage ist nicht sensitiv zur gewünschten Spalte (Geburtsjahrgang)

```
nats18 <181> grep '.*   .*      35' personal
Maier   Klaus   35      A3      7500
Moll    Grit    37      D3      3500
```

→ Regulärer Ausdruck ist nicht verankert

```
nats18 <181> grep '^[^  ]*      [^      ]*      35' personal
Maier   Klaus   35      A3      7500
nats18 <181> grep '.*   .*      35      .*      ' personal
Maier   Klaus   35      A3      7500
```

Arithmetische Intervalle: Approximation durch Regulären Ausdruck
Beispiel: Alle Mitarbeiter, die zwischen 1945 und 1970 geboren wurden

```
nats18 <196> egrep '.*  .*      5[0-9]  .*      |.*     .*
      6[0-5]    .*      .*' personal
Schulze Frieda  53      B2      8100
Heine   Hans    61      C1      4300
```

Probleme:

- ☐ Stärker strukturierte Daten (z.B. Adressen)
- ☐ Verknüpfung mehrerer Tabellen (z.B. persönliche Daten, Abteilungsdaten)