

Beispiel 2 (Mehrdeutigkeit)

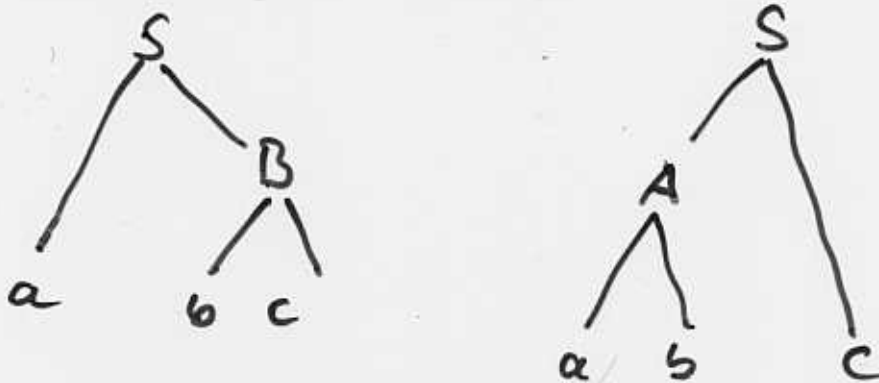
G: $S \rightarrow aB \mid Ac$

$A \rightarrow ab$

$B \rightarrow bc$

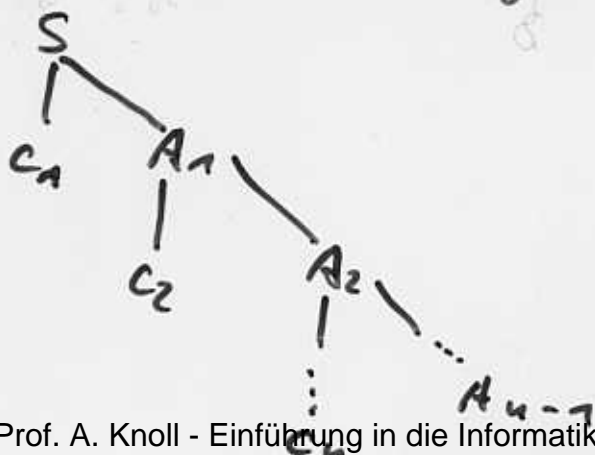
98

Ableitung für das Wort abc:



Zwei unterschiedliche Bäume für diese Ableitung. Hinweis: Bei best. Grammatiken sind solche Mehrdeutigkeiten nicht zu beseitigen

Beispiel 3: (Reguläre Grammatik). Da die Seite ^{rechte} von Regeln regulärer Gr. einzelne Terminale oder Terminal gefolgt von Variablen sind, ergibt sich immer die Form eines entarteten Baums (nach rechts geneigte lineare Liste)



2. $b^*(ab^*ab^*ab^*)^*$ 'alle Wörter über $\{a, b\}$, deren Anzahl an "a" durch 3 teilbar ist. 99
3. $(0|(0|1)^*0)$ 'x = 0 oder x endet mit 0'

Man kann zeigen: die Menge der durch reguläre Ausdrücke beschreibbaren Sprachen ist genau die Menge der Typ-3-Sprachen (Borel's später).

Konkretisierungen:

- MS-DOS: ? steht für genau einen bel. Buchstaben
 * steht für eine beliebige (a. U. auch leere) Buchstabenfolge

Damit beispielsweise über dem Alphabet $A = \{a, b, c\}$ folgende Ausdruck möglich:
 $abc? a * bbc$; mit unseren Ausdrucksmitteln für RA lautet das:

$$abc(a|b|c)a(a|b|c)^*bbc$$

UNIX-grep (egrep, fgrep) sowie Derivate und Erweiterungen, z. B. in PERL
 ↳ Global regular expression printer
 ↳ über Perl und Kornschale hinweg

- a) Wild cards :
 - ein bel. Zeichen
 - # eine bel. Zahl
 - ^ Zeilenanfang
 - \$ Zeilenende

b) Mengen und Intervalle:

- [x y z] Ein bel. Teil der Menge {x, y, z}
- [^ x y z] Ein bel. Zeichen auf x, y, z
- [a - b] " im Bereich "a bis b"

c) Escape "\". Wird benutzt, um nicht druckbare Zeichen zu spezifizieren.

- \t Tab, \r Wagenrücklauf, \n Zeilenvorschub, \f Seitenwechsel oder
- \a ist a in Anstrichen, die a als Meta-Zeichen enthalten.

d) Wiederholungen

- y^* Null oder beliebig viele y
- y^+ Ein oder beliebig viele y
- $y^?$ Null oder ein y

Beispiele:

- '.*\?[^&]*&.*' trifft z.B.
- '/example.t?a=1&b=2&c=3'
- './.*\?cgi' trifft z.B.

'/users/knoll/admin.cgi'

oder mit

'/admin.cgi'

'^From: Knoll'

Jede Zeile mit diesem
Anfang wird gefunden

'\\$?([0-9,]+)\.(\#\#)' trifft z.B.

{
1,234.56
\$4,296,458.19
\$3,5,6,4.000

B) Endliche Automaten

Allgemein: universelle Modellierungswerkzeuge
für dynamische Abläufe (Zustand führt
auf Zustand). Verwendung in Informatik
um Symbol- oder Handlungsfolgen zu
überprüfen, zu analysieren oder abzusuchen.
Für die meisten aus dem tägl. Leben be-
kannten Umgangssprachl. als 'Automaten'
bezeichneten Geräte existieren Modelle in
Form von EA. Beispiel: Getränke auto-
mat, Getränke automat, Geld automat.

Wir betrachten EA als Akzeptoren für
Typ 3-Sprachen und als Modelle für getaktete
technische Systeme.

- Veranschaulichung des Automaten über gerichtete Graphen: ~~Knoten~~ Knoten symbolisieren Zustände; Kanten tragen ein Symbol aus dem Arbeitsalphabet A , beim Lesen des betreffenden Zeichens wird ~~ein~~ Zustands-~~übergang~~ der betreffende Zustandsübergang vollzogen.
- Der Automat hat ausgezeichnete Zustände: ein Startzustand S und mindestens ein Endzustand E
- Verarbeitung eines Worts $w \in A^*$ dadurch, daß Automat bei S anfängt, Symbol für Symbol aus dem Wort konsumiert wird und dabei die entsprechenden Zustandsübergänge vollzogen werden. Sobald ein Endzustand E erreicht, gilt w als akzeptiert (ist also ~~das~~ Element der durch den Automaten M definierten Sprache $L(M)$; auch als $T(M)$ bezeichnet)

'/users/knoll/admin.cgi'

oder mit

'/admin.cgi'

'^From: Knoll'

Jede Zeile mit diesem
Anfang wird gefunden

'\\$?([0-9,]+)\.(\#\#)' trifft z.B.

{
1,234.56
\$4,296,458.19
\$3,5,6,4.000

B) Endliche Automaten

Allgemein: universelle Modellierungswerkzeuge
für dynamische Abläufe (Zustand führt
auf Zustand). Verwendung in Informatik
um Symbol- oder Handlungsfolgen zu
überprüfen, zu analysieren oder abzusuchen.
Für die meisten aus dem tägl. Leben be-
kannten Umgangssprachl. als 'Automaten'
bezeichneten Geräte existieren Modelle in
Form von EA. Beispiel: Getränke auto-
mat, Getränke automat, Geld automat.

Wir betrachten EA als Akzeptoren für
Typ 3-Sprachen und als Modelle für getaktete
technische Systeme.

- Veranschaulichung des Automaten über gerichtete Graphen: ~~Knoten~~ Knoten symbolisieren Zustände; Kanten tragen ein Symbol aus dem Arbeitsalphabet A , beim Lesen des betreffenden Zeichens wird ~~ein~~ Zustands-~~über~~ der betreffende Zustandsübergang vollzogen.
- Der Automat hat ausgezeichnete Zustände: ein Startzustand S und mindestens ein Endzustand E
- Verarbeitung eines Worts $w \in A^*$ dadurch, daß Automat bei S anfängt, Symbol für Symbol aus dem Wort konsumiert wird und dabei die entsprechenden Zustandsübergänge vollzogen werden. Sobald ein Endzustand E erreicht, gilt w als akzeptiert (ist also ~~das~~ Element der durch den Automaten M definierten Sprache $L(M)$; auch als $T(M)$ bezeichnet)