

# **Mustererkennung mit Baumautomaten**

Eine Ausarbeitung von Gisse Alvarado

für das Seminar

**„Mustererkennung mit syntaktischen und graphbasierten Methoden“**

bei Prof. Dr. W. Kurth/ Th. Mangoldt

Cottbus 2006

# Inhalt

## **1. Baumautomaten**

1.1. Definition

1.2. Verfahren

1.3. Erkannte Sprache

1.4. Konstruktion eines Baumautomaten aus einer Baumgrammatik

## **2. Ein einfaches Beispiel**

2.1. Baumgrammatik

2.2. Stufenweise Bearbeitung durch den Baumautomaten

## **3. Ein komplexes Beispiel**

3.1. Fingerabdruck-System

3.2. Operationen

3.3 Beispiel

# 1. Baumautomaten

## 1.1. Definition

- Allgemein : endlicher Automat, der auf Bäumen operiert
- Methode : „frontier-to-root“  $\approx$  „von\_der\_Grenze\_zur\_Wurzel“  
→ Start bei den Blattknoten, dann geht man aufwärts zur  
Wurzel
- System :  $\mathcal{A} = (Q, F, \{f_a \mid a \text{ in } \Sigma\})$   
wo :
  - Q = endliche Menge von Zuständen der Knoten des Baums,
  - F = Menge von Endzuständen, und auch Teilmenge von Q,
  - $f_a$  = eine Relation  $Q^n \times Q$ , wobei n die Stelligkeit von a ist,
  - $\Sigma$  = Alphabet
- $\mathcal{A}$  ist eine nichtdeterministische Maschine

## **1.2. Verfahren**

- Zuerst : Zu jedem Blattknoten wird ein Startzustand festgesetzt.
- Dann : Pfade werden gleichzeitig aufwärts zur Wurzel abgesucht.
- Danach :  $\mathcal{A}$  weist einen Zustand für jeden Knoten anhand der Relationen  $f_a$  zu.
- Schließlich:  $\mathcal{A}$  akzeptiert einen eingegebenen Baum, wenn  $\mathcal{A}$  am Schluss in einen Endzustand eintreten kann.

## **1.3. Die erkannte Sprache**

- $\mathcal{A}$  erkennt folgende Sprache :

$$T(\mathcal{A}) = \{T \in T_\Sigma \mid \mathcal{A} \text{ kann in einem Endzustand anhalten, wenn die Wurzel von } T \text{ erreicht ist}\}$$

## 1.4. Baumautomat aus einer Baumgrammatik

➤ Sei gegeben:

- \* Baumgrammatik  $G_t = (V, r, S, P)$ , die die Menge von Bäumen  $L(G_t)$  generiert,  $V = N \cup T$

➤ Gesucht :

- \* Baumautomat  $\mathcal{A}$ , der  $L(G_t)$  erkennt

➤ Lösung:

- \* Wir setzen  $Q = N$  mit  $F = \{S\}$ ,
- \* dann wird für jedes Symbol  $a$  aus  $\Sigma$  eine Relation  $f_a = (X_1, \dots, X_n, X)$  definiert,
- \* wenn es eine solche Produktion in  $G_t$  gibt:

$$\begin{array}{c} X \rightarrow a \\ \quad / \quad \backslash \\ X_1 \dots X_n \end{array}$$

## 2. Ein einfaches Beispiel

### 2.1. Baumgrammatik

➤ Gegeben :

eine expansive Baumgrammatik  $G_t = (\{S, A, x, y, z, \$\}, r, S, P)$ ,

wo:

- \* Nichtterminale : S, A
- \* Terminale : x, y, z, \$
- \* Produktionen :

$$A \rightarrow x$$

$$A \rightarrow y$$

$$A \rightarrow z$$

$$S \rightarrow \$$$

$$\begin{array}{c} | \\ A \end{array}$$

$$\begin{array}{c} / \ \backslash \\ A \quad A \end{array}$$

➤ Der Automat  $\mathcal{A}$  wird spezifiziert als :

$$\mathcal{A} = (\{S, A\}, \{S\}, \{f_x, f_y, f_z, f_\$\},$$

mit:

$f_x = \{(\emptyset, A), \text{erzeugt aus der Produktion } A \rightarrow x\}$

$f_y = \{(\emptyset, A), \text{erzeugt aus der Produktion } A \rightarrow y\}$

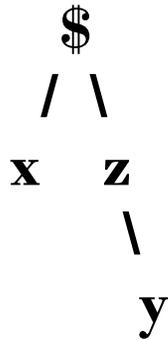
$f_z = \{(A, A), \text{erzeugt aus der Produktion } A \rightarrow z\}$

|  
A

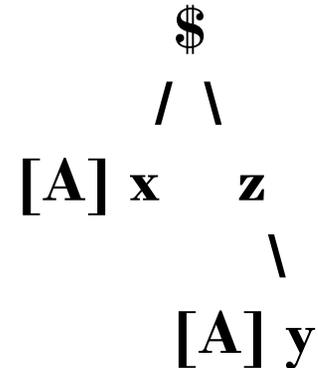
$f_\$ = \{(A, A, S), \text{erzeugt aus der Produktion } S \rightarrow \$\}$

/ \  
A A

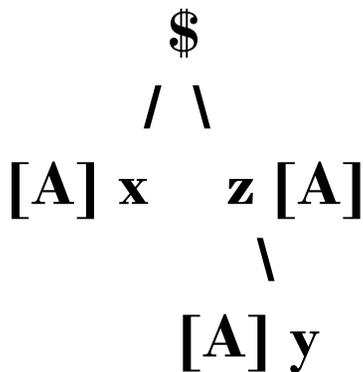
## 2.2. Stufenweise Bearbeitung durch den Baumautomaten



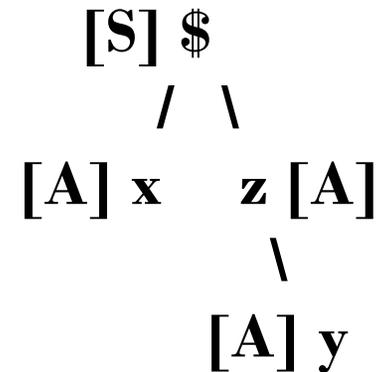
(a) Eingangsbaum



(b) Zustandszuweisung zum Blattknoten



(c) Zustandszuweisung zur  
Nachkommenschaft der Wurzel



(d) Zustandszuweisung an die Wurzel

## 3. Ein komplexeres Beispiel

### 3.1. Fingerabdruck- System

- entwickelt von Moayer und Fu (1974, 1976)
- Aufgabe : Fingerabdruckerkenung mittels Baumgrammatiken und Automaten
- Ziel :
  - \* Verbesserung des manuellen Verfahrens
  - \* Klassifizierung des Fingerabdrucks
- Dieses System benutzt eine sorgfältige Kombination von Algorithmen

- Eigenschaft eines Fingerabdrucks wird beschrieben durch „*minutiae*“
  - dient als primitives Muster
  - und ist die essentielle Eigenschaft einer Fingerabdruck-Linie
  - abruptes Ende, Verzweigungen, Punkte, Schnitte, abgetrennte Teile
- Es gibt ungefähr 50 bis 100 minutiae pro Fingerabdruck
  - für die Identifizierung genügt weniger als ein Dutzend
- Angenommen : 50 minutiae kommen vor
  - es gibt ungefähr  $2^{50}$  oder  $10^{15}$  verschiedene Fingerabdrücke
- Arten und Stellen der „*minutiae*“ sind gut gemischt, so dass verhindert wird, dass der gleiche Fingerabdruck bei mehreren Individuen vorkommt.

## **3.2.Operationen**

- Digitalisierung und Segmentierung der Abbildung eines Fingerabdrucks
- Bildverbesserung
- Vorverarbeitung
- Linienverfolgung (Tracing) und Linienkodierung
- Nachverarbeitung
- Merkmalsextraktion
- **Darstellung als Baum**
- Einordnung in eine grundlegende Klasse

➤ **Digitalisierung und Segmentierung der Abbildung eines Fingerabdrucks:**

- \* Das Bild wird gescannt,
- \* dann wird es in eine  $256 \times 256$  Matrix konvertiert,
- \* jedem Matrixpunkt wird ein Grauwert in der Auswahl von 0 bis 255 zugewiesen,
- \* Bildränder werden entfernt, weil sie weniger oder gar keine Informationen geben,
- \* Um Kompliziertheit zu reduzieren, wird das Bild in eine  $4 \times 4$  Matrix von Fenstern zerlegt,
- \* jedes Fenster ist eine Matrix von  $48 \times 48$  Punkten

→ Verschiebung der Orientierung eines Musters verursacht eine verschiedene Klassifizierung des Fensters.

## Typisches Muster eines Fingerabdrucks



4.24. Whorl pattern fingerprint. (From Moayer and Fu [1976b].)

➤ **Bilderverbesserung:**

\* Wiederholte Verwendung des Laplace-Operators:

$$\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$$

\* jedem Punkt entspricht einer gewichteten Summe der Graustufen in einem Bereich.

\* Graustufenumwandlung  $f_{ij}$ , wobei:  $i, j$  sind Koordinaten:

$$f'_{ij} = 1.92 f_{ij} + 0.154(f_{i+3,j} + f_{i-3,j} + f_{i+2,j+2} + f_{i+2,j-2} + f_{i-2,j+2} + f_{i-2,j-2})$$

\* Justierungsfaktoren 1.92 und 0,154 wurden nach vielen Versuchen bestimmt.

\*Die abschließende Binärdarstellung  $f_{ij}$ :

$$f_{ij}'' = \begin{cases} \mathbf{1} & \text{if } f_{ij}' \geq t_1 \\ \mathbf{0} & \text{if } f_{ij}' \leq t_2 \end{cases}$$

wo:  $t_1(n)$  und  $t_2(n)$  sind abhängig von der Wiederholungszahl  $n$  und beeinflussen die Konvergenz zum Binärbild

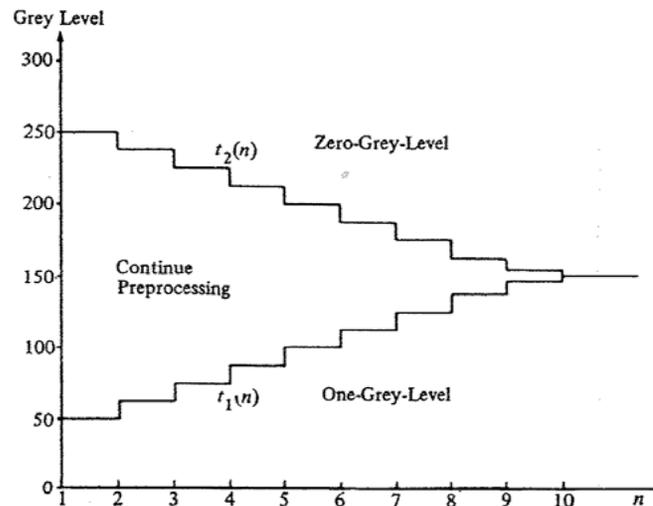


Figure 4.23. Convergence function for repetitions of the Laplacian operator. (From Moayer and Fu [1976b].)



## ➤ Vorverarbeitung, Linienverfolgung und Linienkodierung

### Funktionen:

- \* Binärfenster absuchen, um Linien zu finden,
- \* dann werden die Variationen der Liniendicke unterdrückt,
- \* und es wird eine Verschlüsselung der Linienskelette anhand eines Richtungscode erzeugt.

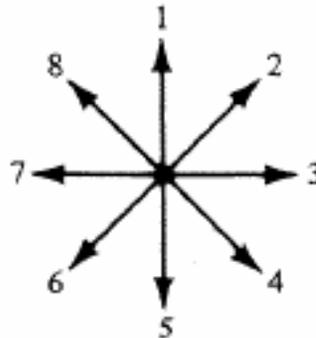


Figure 4.26. Directional code.

## **Vorverarbeitung:**

- \* Suche nach überlappenden benachbarten Linien, um falsche Primitive zu vermeiden

## **Linienverfolgung und Linienkodierung:**

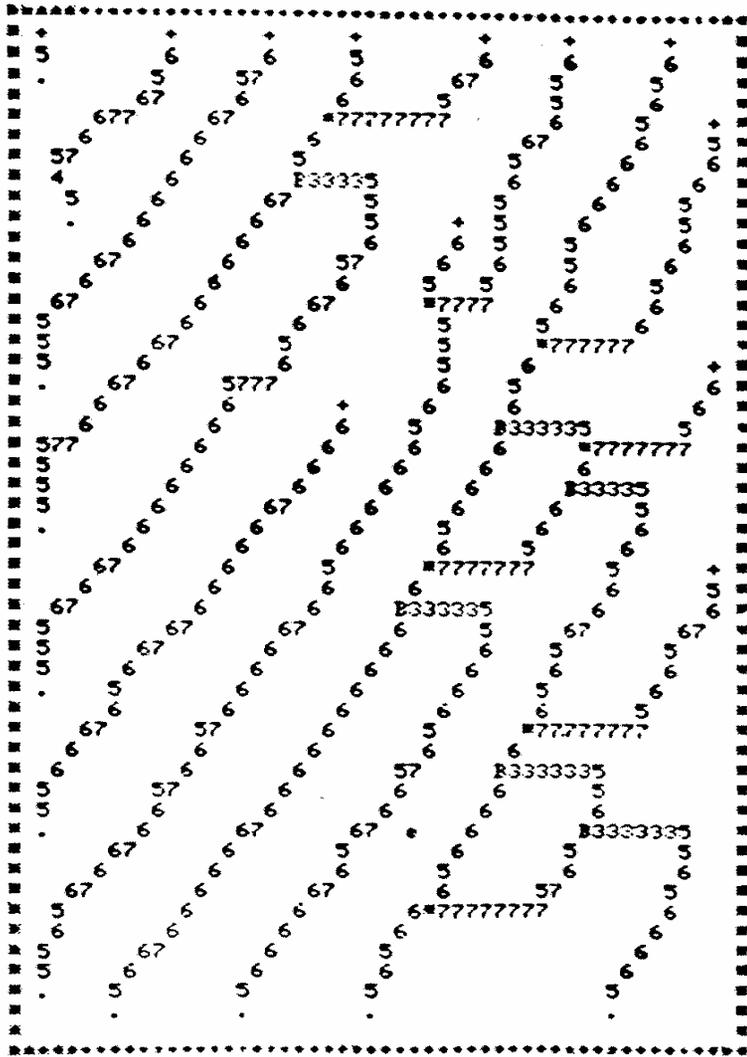
- \* Geht aus von skelettartigem Bild
- \* Anwendung des Richtungscode,
- \* Erzeugung des Kettencodes für jede Linie

→ Kettenverschlüsselte Darstellung, in der der Richtungscode mit den Symbolen +, •, \* und B erscheint.

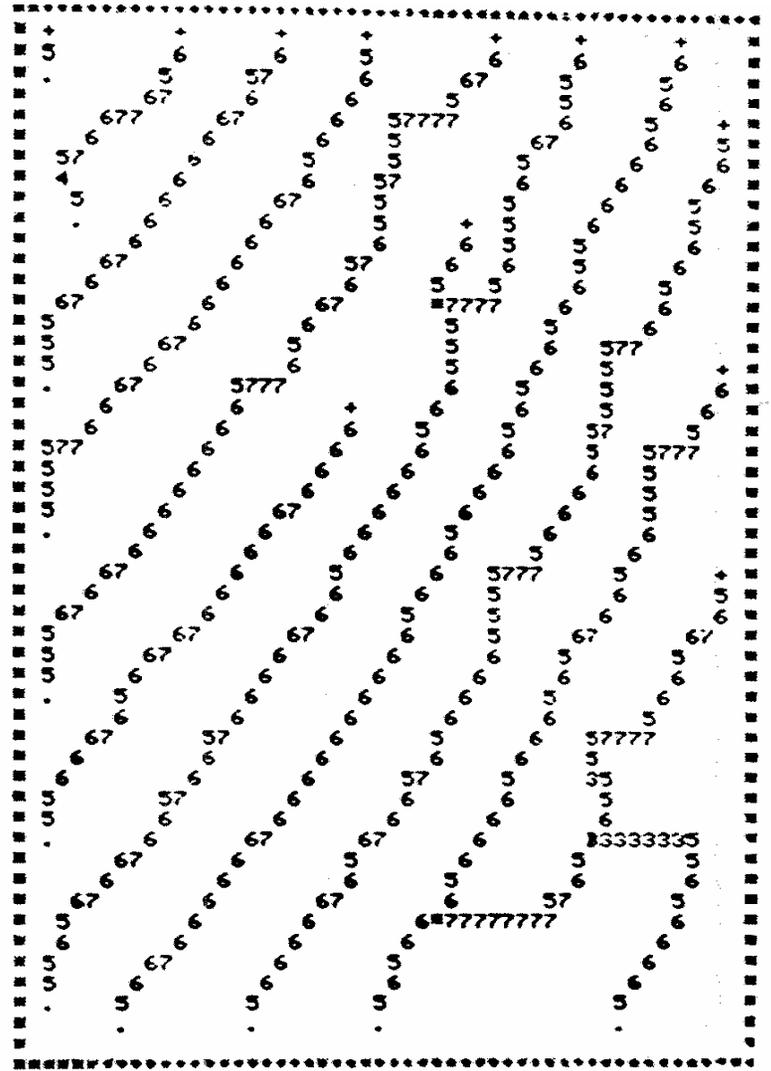
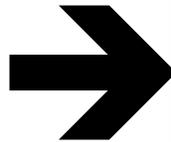


## ➤ **Nachbearbeitung**

- \* kritische Phase überhaupt
- \* versucht, weitere falsche Primitive zu entfernen
- \* Untersuchen der richtungscodierten Linien
- \* Nachverarbeitungsroutine :
  1. Linien sind ununterbrochen;
  2. Linien laufen im allgemeinen parallel;
  3. Der Anfang oder das Ende einer Linie befindet sich im allg. zwischen 2 Linien;
  4. Variation und Krümmung ändern sich allmählich  
(außer in der Umgebung von Deltas, von Zentren von Schleifen,  
von Zentren von Wirbeln)



Linienverschlüsselung



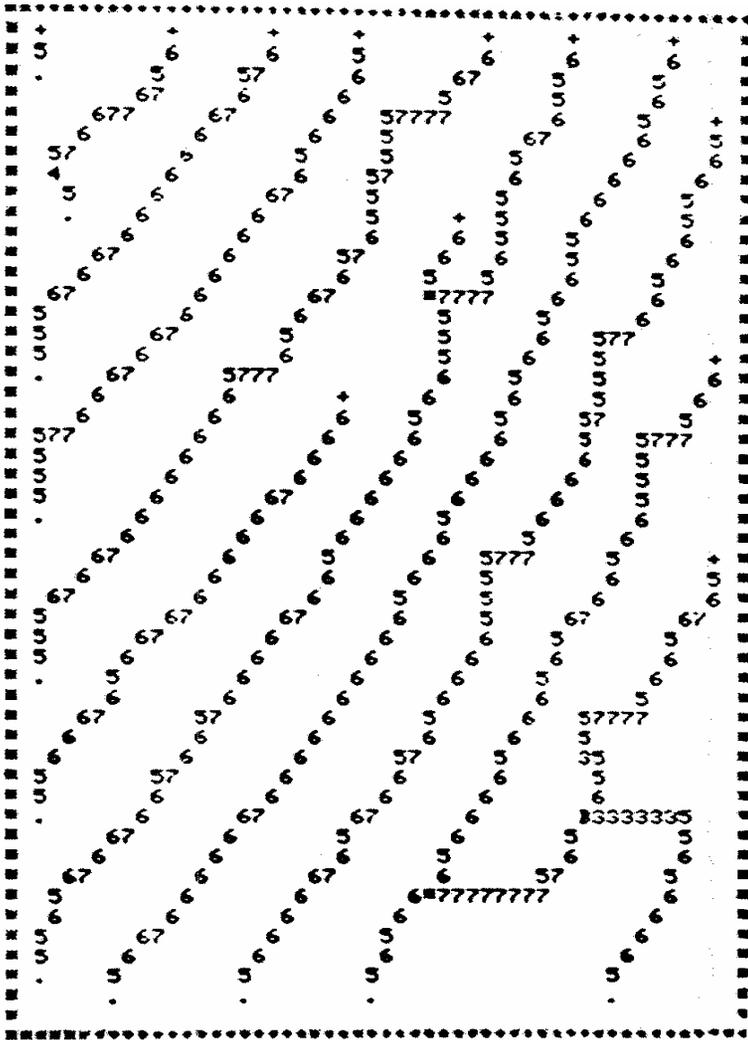
Linien für die Weiterverarbeitung

## ➤ Merkmalextraktionsphase:

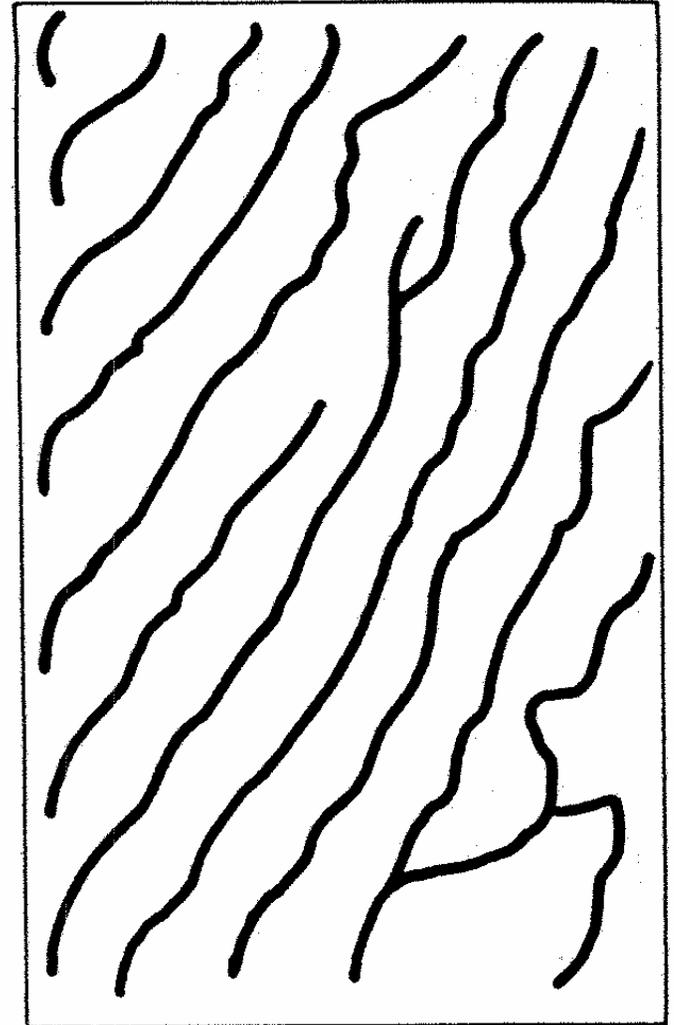
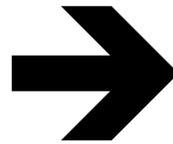
- \* Identifizierung der endgültigen „minutiae“, je nach Art und Lage
- \* Zu beachten ist:
  1. Es gibt ununterbrochene Linien von einem Fenster zu anderen.
  2. Es gibt Linienabschlüsse, Verzweigungen, Verschmelzungen und „Segmente“.
  3. Es gibt Informationen in den relativen Positionen der Linien zu einander.
- \* L und R geben Informationen über die Lage der Linie.
- \* „ein abruptes Ende“ einer Linie wird identifiziert, wenn ihre Koordinaten  $i, j$  die
$$3 \leq i \leq 48 \quad \text{und} \quad 3 \leq j \leq 48$$
zufrieden stellen

# Primitives Muster eines Fingerabdrucks

Terminal	Interpretation	Description
A	(	Directional Code
B	)	Directional Code
C	⌒	Directional Code
D	⌒	Directional Code
X		Branching Point
*		Merging Point
-		Loop
\$		Sudden Ending
\$-\$		Segment
L	Left Side	Left
R	Right Side	Right
●	Starting Point	Starting Ridge



Input Linien



Darstellung als primitives  
Muster

## ➤ Darstellung als Baum:

- \* Satz von Knoten in der Form einer dreifach verbundenen Liste
- \* Jeder Knoten enthält 4 Felder :  
Symbol, B, S und C
- \* Felderbeschreibung:
  - ✧ Symbol-Feld enthalten die Primitive
  - ✧ B-Feld sind ein Zeiger zum vorhergehenden Knoten im Baum.
  - ✧ S-Felder verbinden zur nächst benachbarten Linie.
  - ✧ C-Feld sind ein Zeiger zum Knoten, der die nächsten Primitive für die Linie der unteren Ebenen enthält.
- \* Diese Knotenbeschreibung wird automatisch von einer Routine aus dem Fenster genommen.



## ➤ **Einordnung in eine grundlegende Klasse**

- \* Erkennung der Baumdarstellung aus Fenstern
- \* Jeder Klasse wird eine expansive Baumgrammatik zugeordnet,
- \* und ein Fenster muss in die Klasse der Sprache, deren Grammatik seine Baumdarstellung erzeugt, eingeordnet werden.



\* Automat  $\mathcal{A} = (\{S, A\}, \{S\}, \{f_x, f_y, f_z, f_\$ \})$ ,  
 mit Zustandsmenge  $Q = \{S, T1, T2, T3\}$   
 Endzustand  $F = \{S\}$   
 Relationen :

$$f_\bullet = \{(T3, S)\},$$

$$f_s = \{(T1, T2)\},$$

$$f_R = \{(T2, T3), (T1, T2, T3)\},$$

$$f_{Rf_C} = f_D = \{(\emptyset, T3), (\emptyset, T1), (T1, T1)\}.$$

### 3.3. Beispiel

- Eine der Grundklassen besteht aus solchen Fenstern, die ununterbrochene Linien enthalten zusammen mit *einer* Linie mit *einer* Verzweigungs-minutiae.
- eine expansive Baumgrammatik dieser Klasse ist :

$$G_t = (V, r, S, P)$$

mit :

$$N = \{N1, N2, N3, N4, N5, N6, N7\};$$

$$\Sigma = \{\bullet, X, A, B, C, D, R, L\};$$

S = einzelnes Nichtterminal, N1;

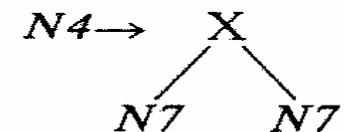
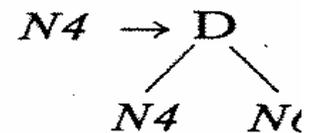
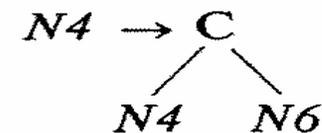
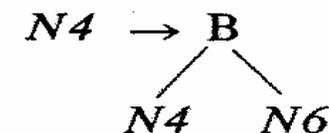
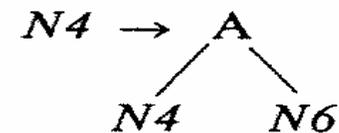
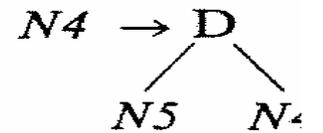
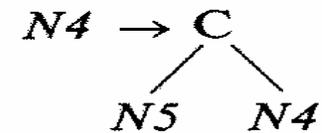
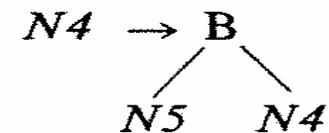
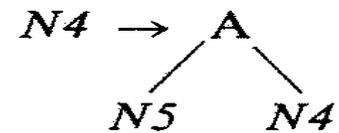
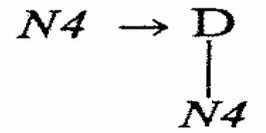
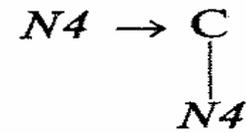
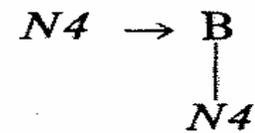
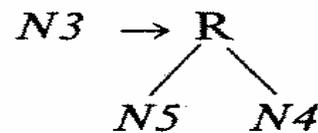
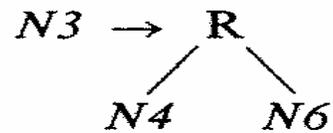
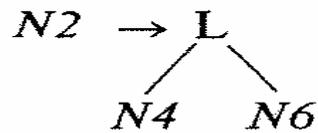
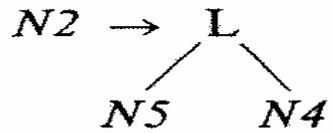
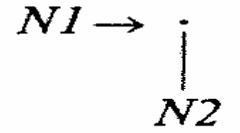
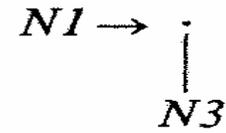
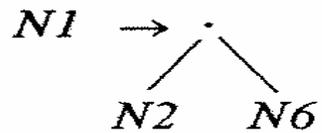
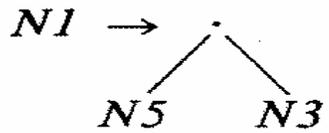
Stelligkeiten der Terminale :

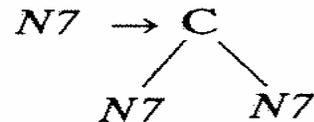
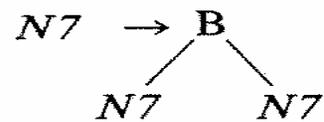
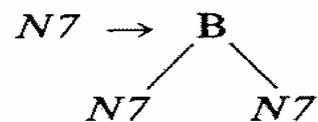
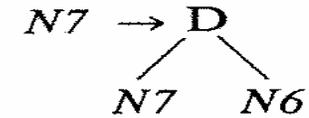
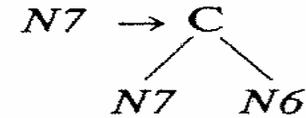
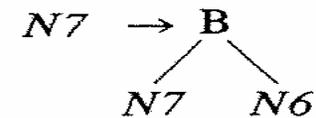
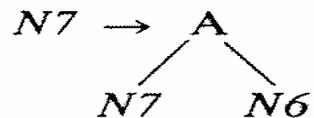
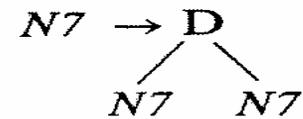
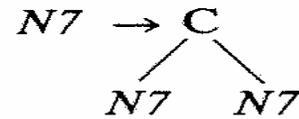
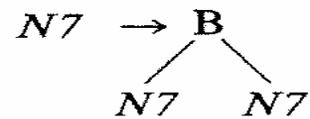
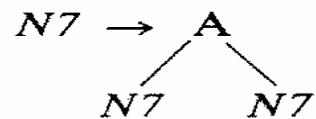
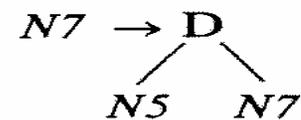
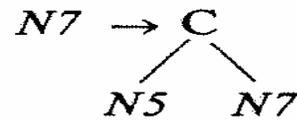
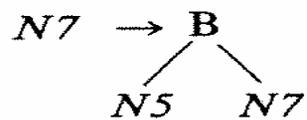
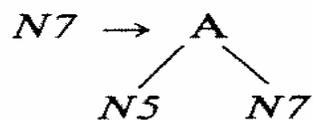
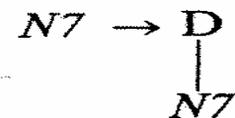
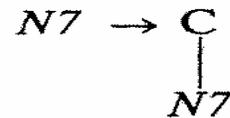
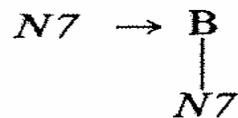
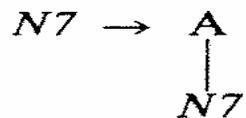
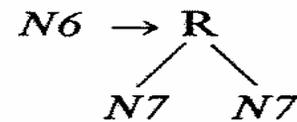
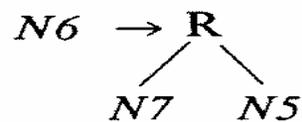
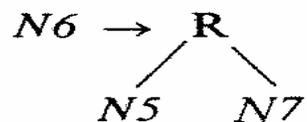
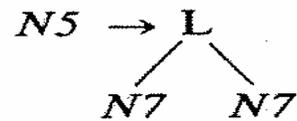
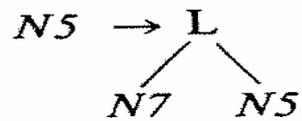
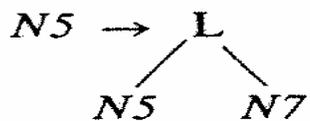
$$r(R) = r(L) = r(X) = \{2\},$$

$$r(\bullet) = \{1, 2\},$$

$$r(A) = r(B) = r(C) = r(D) = \{0, 1, 2\};$$

# Produktionen:





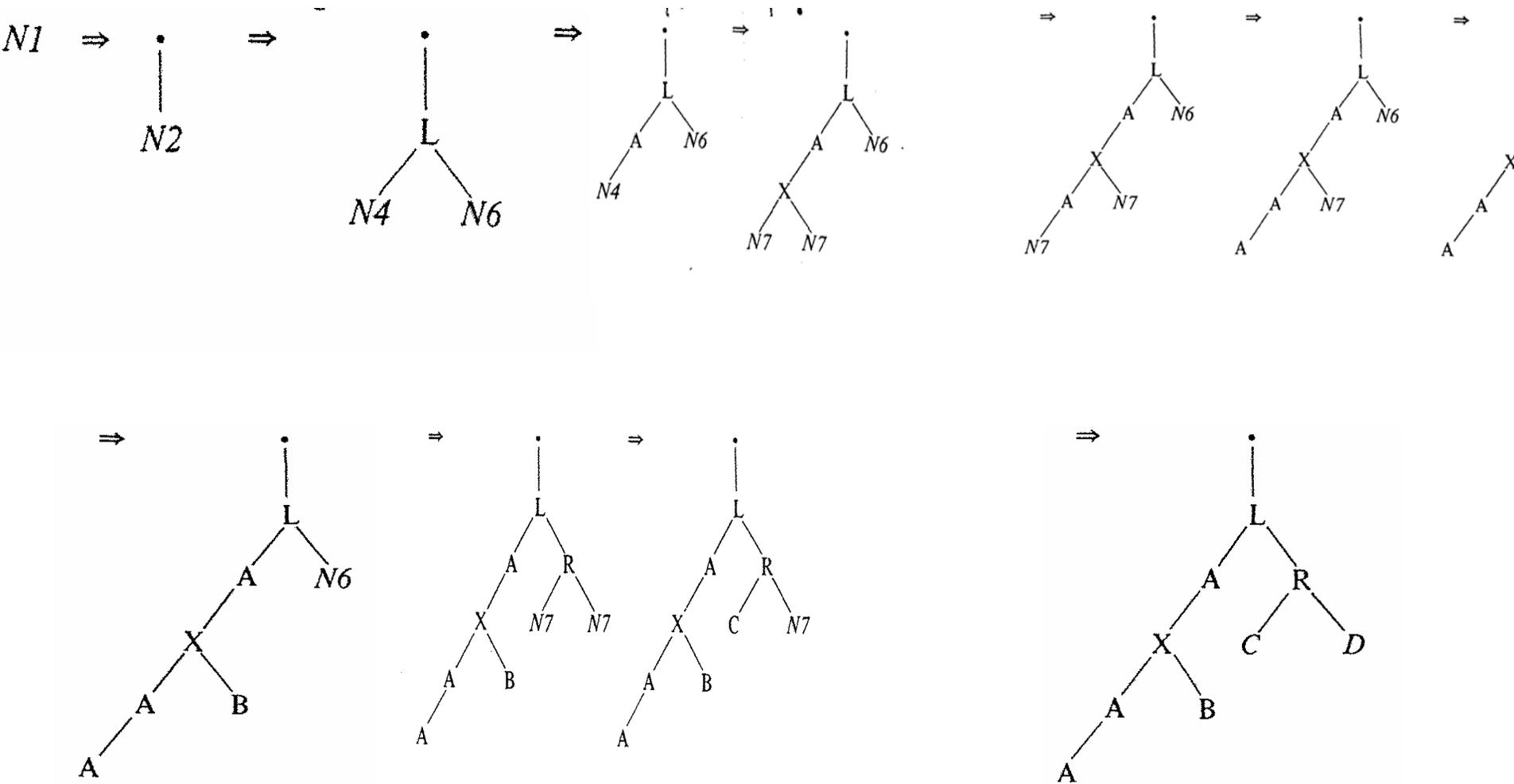
$N7 \rightarrow A$

$N7 \rightarrow B$

$N7 \rightarrow C$

$N7 \rightarrow D$

# Die von $G_t$ benutzten Ableitungen :



Das resultierende Muster von einem Fingerabdruck mit Hilfe der Baumgrammatik ist:

