

Seminar “Mustererkennung mit syntaktischen und graphbasierten Methoden”

Prof. Dr. W. Kurth / Th. Mangold

Wintersemester 2006/2007

Thema 14

Formelerkennung mit Baum-Transformationen

Präsentiert von Marcin Fabisz

Gliederung:

1. Einführung und Motivation
 - 1.1. Herausforderungen
 - 1.2. Mathematikererkennung mit Baumtransformation
2. Sybollayout in Formeln
 - 2.1. Operatordominanz und Basislinien
 - 2.2. Basislinienstrukturbäume (BST)
 - 2.3. Symbolklassen
3. Layout-Durchgang
 - 3.1. Vorbereitung der Eingabe
 - 3.2. Hauptschritte
 - 3.3. Hauptfunktionen
4. Lexikalische Analyse
 - 4.1. Komplexe Symbole
 - 4.2. Strukturierte Symbole
5. Formelanalyse
6. Testergebnisse und Zusammenfassung
 - 6.1. DRACULAE-Tests mit UW-III
 - 6.2. DRACULAE-Tests mit FFES
7. Quellen

1. Einführung und Motivation

- Forschungsthema seit 30 Jahren.
- Konvertierung gedruckter, wissenschaftlicher Arbeiten in elektronische Form zur Bearbeitung oder Audiowiedergabe für Blinde.
- Erkennung handgeschriebener Formeln von Scans, PDAs oder Tablets als Alternative zur Eingabe mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ oder Formel-Editoren.

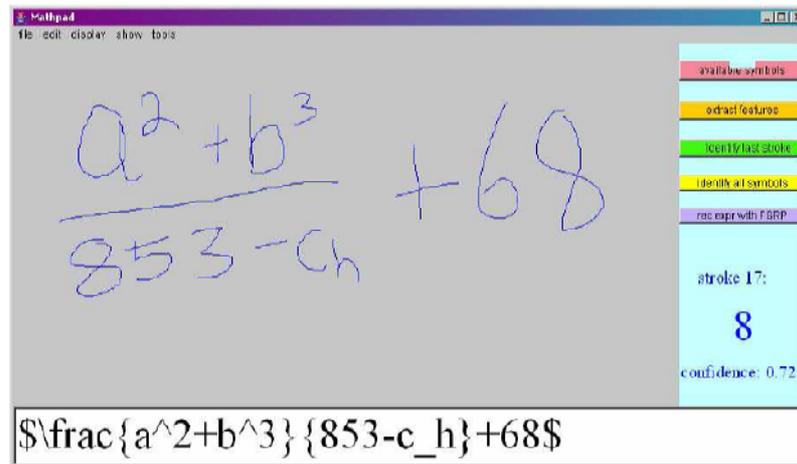


Abbildung 1: Konvertierung Handschrift zur $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, Quelle: [1]

1.1. Herausforderungen

- Formelsuche im Dokument, das Text, Ausdrücke und Graphiken beinhalten kann.
- Formeln können in der Zeile oder versetzt sein.
- Große Anzahl von Symbolen, Schriftarten, Schriftstilen und Schriftgrößen wird benutzt.
- Unterschied zwischen Rauschen und kleinen Symbolen wie Kommas und Punkten soll erkannt werden.
- Erkennung räumlicher Relationen zwischen Symbolen besonders bei Handschrift.
- Mehrdeutigkeiten im Layout.
- Mehrdeutige Rolle verschiedener mathematischer Symbole benötigt kontextuelle Analyse.
- Wenig Redundanz als Hilfe bei Mehrdeutigkeiten.
- Mehrzahl von mathematischen Dialekten.

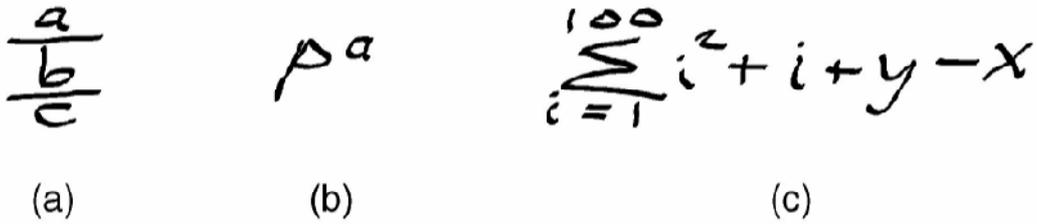


Abbildung 2: Mehrdeutiger Layout, Quelle: [2]

1.2. Mathematikerkennung mit Baumtransformation

- Suche nach linearen Strukturen – Basislinien.
- Sekundäre Basislinien.
- Symbolspezifische Unterbereiche.
- Links-Nach-Rechts-Analyse ohne Backtracking.
- Die Basislinien-Struktur wird als Basislinien-Strukturbaum (*Baseline Structure Tree* - BST) organisiert.
- 3 Phasen der Bearbeitung:
 - Layout Durchgang ohne Fehleranalyse mit entsprechenden Symboldefinitionen generiert BST.
 - Lexikalischer Durchgang liefert das *Lexed BST*.
 - Formelanalyse-Durchgang liefert den Operatorbaum.
- Diagrammerkennungssapplikation für Rechnerisches Verstehen Großer Algebraischer Ausdrücke (*Diagram Recognition Application for Computer Understanding of Large Algebraic Expressions* – DRACULAE).
- Transformationsregeln sind in einer speziellen ASCII-Sprache implementiert; Baumtransformationssprache TXL.

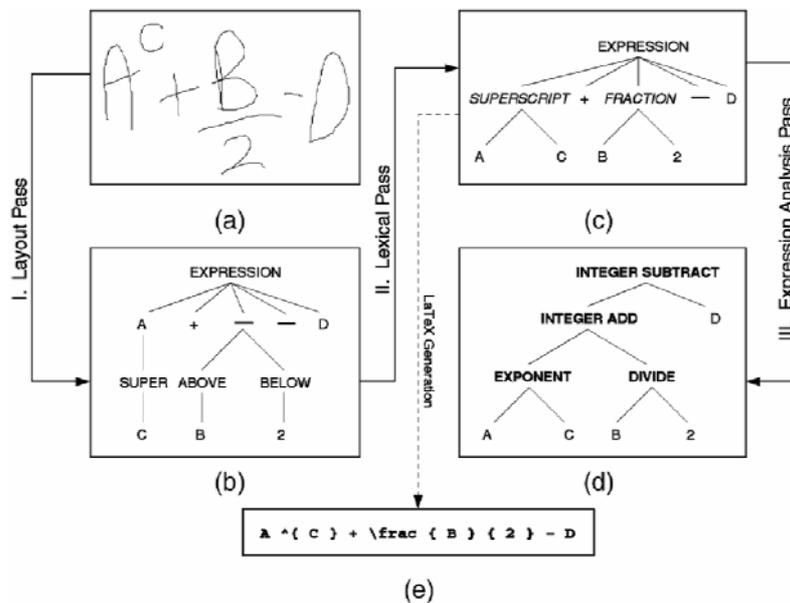
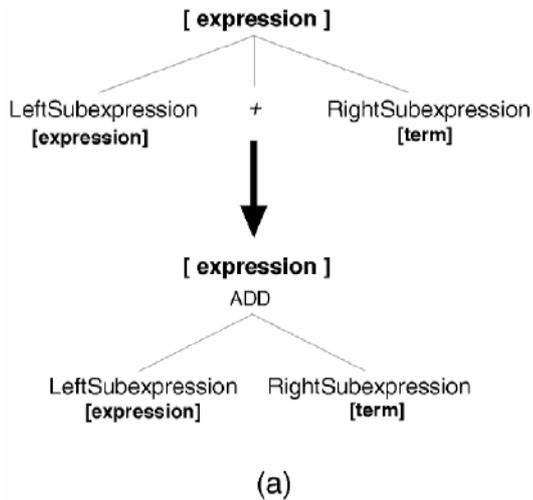


Abbildung 3: Überblick von DRACULAE, Quelle: [2]



```

rule convertAdditionsToOperatorTrees
  replace [expression]
    LeftSubexpression[expression] +
    RightSubexpression[term]
  by
    ADD LeftSubexpression
    RightSubexpression
end rule

```

(b)

Abbildung 4: Transformationsregel im TXL-Code, Quelle: [2]

2. Sybollayout in Formeln

2.1. Operatordominanz und Basislinien

- Reichweite des Operators.
- Dominanz – Operator A dominiert Operator B, wenn B ist in Reichweite von A und A ist nicht in Reichweite von B.
- Basislinie.
- Verschachtelte Basislinie.
- Dominante Basislinie.
- Startsymbol.

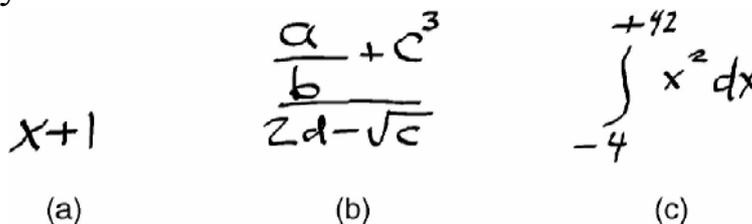


Abbildung 5: Startsymbolbeispiele, Quelle: [2]

2.2. Basislinienstrukturbäume (BST)

- Symbol-Knoten.
- Bereich-Knoten.
- Bereich-Kennzeichnung. *ABOVE, BELOW, SUPER, SUBSC, UPPER, LOWER, TLEFT, BLEFT, CONTAINS, EXPRESSION*

2.3. Symbolklassen.

- Centroid- und Grenzwert-Verhältnis. $c, t \in (0, 0.5), t \leq c$
- Centroid-Bedeutung.
- Überlappende Bereiche. $x^2 \sum_{i=1}^{10,000} i$

Symbol Class	y-centroid	Thresholds			
		BELOW	ABOVE	SUBSC	SUPER
<i>Non-Scripted</i> unary/binary operators and relations (+, -, =, ≥, →, etc.)	$\frac{1}{2}H$	$\frac{1}{2}H$	$\frac{1}{2}H$	–	–
<i>Open Bracket</i> (, {, [cH	$minY$	$maxY$	–	–
<i>Root</i> ($\sqrt{\quad}$)	cH	$minY$	$maxY$	tH	$H - (tH)$
<i>Variable Range</i> $\Sigma, f, \Pi, \cup, \cap$	$\frac{1}{2}H$	tH	$H - (tH)$	tH	$H - (tH)$
<i>Plain: Ascender</i> 0...9, A...Z, b,d,f,h,i,k,l,t, $\Gamma, \Delta, \Theta, \Lambda, \Xi, \Phi, \Psi, \Omega, \delta, \theta, \lambda$	cH	tH	$H - (tH)$	tH	$H - (tH)$
<i>Plain: Descender</i> g,p,q,y, $\gamma, \eta, \mu, \rho, \chi, \psi$	$H - (cH)$	$\frac{1}{2}H + t\frac{1}{2}H$	$H - t\frac{1}{2}H$	$\frac{1}{2}H + t\frac{1}{2}H$	$H - t\frac{1}{2}H$
<i>Plain: Centered</i> All other symbols (including Close Brackets)	$\frac{1}{2}H$	tH	$H - (tH)$	tH	$H - (tH)$

Tabelle 1: Symbolklassen, Quelle: [2]

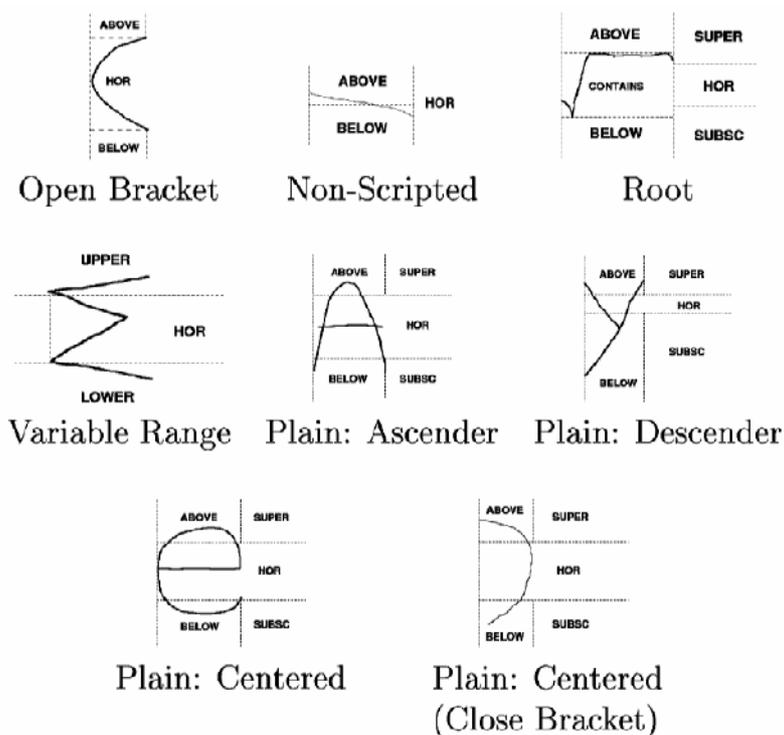


Abbildung 6: Bereiche der Symbolklassen, Quelle: [2]

3. Layout-Durchgang

3.1. Vorbereitung der Eingabe

- Zeichen-Box; $minX(s)$, $minY(s)$, $maxX(s)$, $maxY(s)$
- Symbolklasse; $class(s)$
- Symbol-Centroid; $centroidX(s)$, $centroidY(s)$
- Symbol-Bereiche; $aboveThreshold(s)$, $belowThreshold(s)$, $superThreshold(s)$, $subscThreshold(s)$

3.2. Hauptschritte

- 1) Funktion *BuildBST* erzeugt den Anfangs-BST mit Wurzel *EXPRESSION* und sortierter Symbolliste als Blätter.
- 2) Funktion *Start* findet das Startsymbol in der dominanten Basislinie des Bereichs.
- 3) Die restlichen Symbole in der Basislinie nach dem Startsymbol werden von der Funktion *Hor* erkannt.
- 4) Die Symbole der dominanten Basislinie werden als Kinder des entsprechenden Bereichs ins BST eingefügt.
- 5) Die Symbole der dominanten Basislinie mit ihren eigenen Bereichen teilen den bearbeiteten Bereich in Unterbereiche. Die Symbole, die nicht Teil der dominanten Basislinie sind, werden einem der Unterbereiche zugewiesen.
- 6) Für jeden nicht leeren Unterbereich wird ein Bereichs-Knoten im BST erzeugt und für diesen Unterbereich die Schritte 2 bis 6 rekursiv ausgeführt.

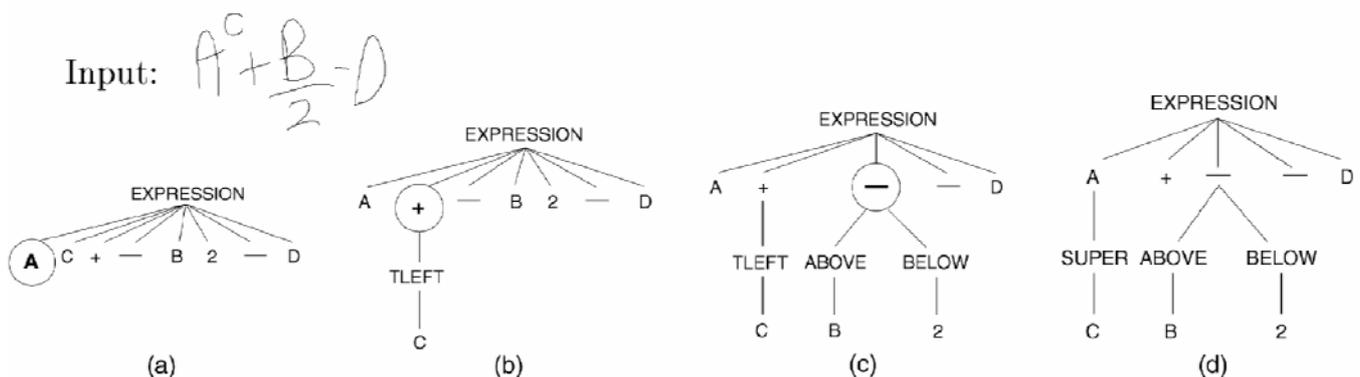


Abbildung 7: Layout-Durchgang, Quelle: [2]

3.3. Hauptfunktionen

- *BuildBST*.
- *ExtractBaseline*.
- *Start*.
- *Hor*; Basisliniensymboldurchlauf und neue Bereiche: *TLEFT*, *BLEFT*, *ABOVE*, *BELOW*, *CONTAINS*.
- *CollectRegions*; Temporäre Bereiche *TLEFT* werden zu *SUPER* oder *UPPER*, bzw. *BLEFT* werden zu *SUBSC* oder *LOWER*.

4. Lexikalische Analyse

4.1. Komplexe Symbole

- Eine Basislinie. Funktionsnamen, Zahlen, Gleichheitssymbol...

4.2. Strukturierte Symbole

- Mehrere Basislinien. Bruch, Grenzen, Summen...

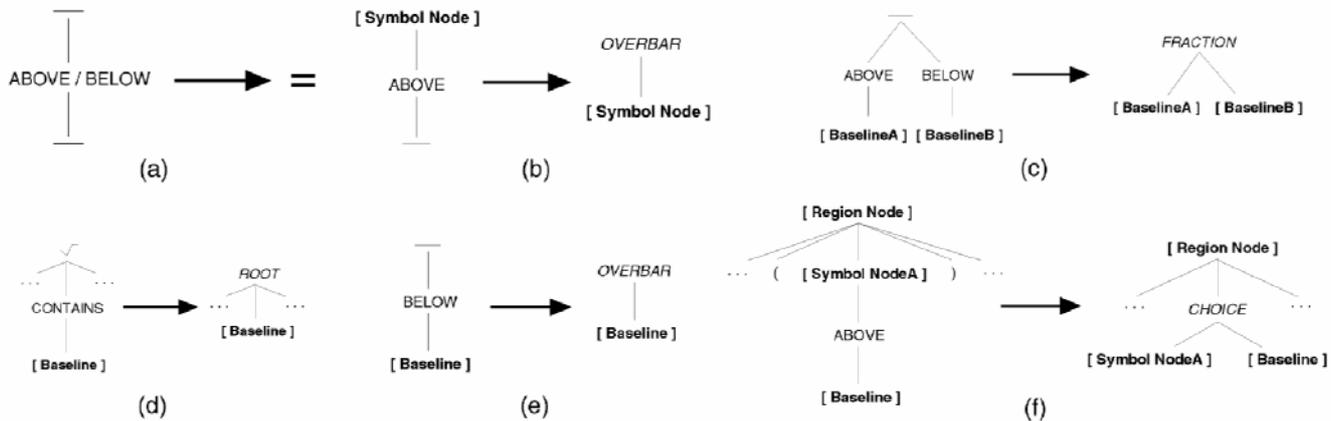


Abbildung 8: Regeln im lexikalischen Durchgang, Quelle: [2]

5. Formelanalyse

- Syntaxanalyse der Formel.
- Semantikanalyse der Formel.

6. Testergebnisse und Zusammenfassung

6.1. DRACULAE-Tests mit UW-III

- *University of Washington English/Technical Document Images Database III (UW-III).*

Threshold Ratio (t)	Centroid Ratio (c)	
	1/3	1/4
1/3	Tokens Correct: 1642 (86%) Baselines: Correct: 463 (71%) Incorrect: 251 Expressions Correct: 20 (27%)	
1/4	Tokens Correct: 1728 (90%) Baselines: Correct: 513 (79%) Incorrect: 165 Expressions Correct: 28 (38%)	Tokens Correct: 1728 (90%) Baselines: Correct: 513 (79%) Incorrect: 165 Expressions Correct: 28 (38%)
1/6	Tokens Correct: 1679 (87%) Baselines: Correct: 471 (79%) Incorrect: 146 Expressions Correct: 27 (37%)	Tokens Correct: 1679 (87%) Baselines: Correct: 471 (79%) Incorrect: 146 Expressions Correct: 27 (37%)
1/8	Tokens Correct: 1679 (87%) Baselines: Correct: 471 (79%) Incorrect: 146 Expressions Correct: 27 (37%)	Tokens Correct: 1658 (86%) Baselines: Correct: 453 (70%) Incorrect: 155 Expressions Correct: 27 (37%)

Tabelle 2: DRACULAE-Tests mit UW-III, Quelle: [2]

6.2. DRACULAE-Tests mit FFES

- Freihand-Formeleingabesystem (*Freehand Formula Entry System* - FFES)

	Input Expression	L ^A T _E X (Lexed BST)	Operator Tree
(a)		$(\frac{\sqrt{ab+b}}{3})^2$	
(b)		$\bar{a} \vee \bar{b} \vee \bar{c}$	(Outside Dialect)
(c)		$\sum_{i=100}^{7426} i + \cos n$	
(d)		$\sum_{i=1}^2 \sum_{j=2}^2 ij$	(Outside Dialect)
(e)		$\frac{\sqrt{\frac{4x^2}{2a}}}{(\int_{2b}^3 \frac{x^e}{4+a} dx)^3} x^2 + \frac{-b + \sqrt{b^2 - 4ac}}{2a}$	(Outside Dialect)

Tabelle 3: DRACULAE-Tests mit FFES, Quelle: [2]

7. Quellen

- [1] J. A. Fitzgerald, F. Geiselbrechtner, and T. Kechadi, "Structural Analysis of Handwritten Mathematical Expressions Through Fuzzy Parsing", The IASTED International Conference on Advances in Computer Science and Technology (ACST 2006), Puerto Vallarta, Mexico, pp. 151-156, 2006; <http://www.csi.ucd.ie/UserFiles/publications/1133283209625.pdf>
- [2] Richard Zanibbi, Dorothea Blostein, James R. Cordy, "Recognizing Mathematical Expressions Using Tree Transformation", IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (11) (2002), S. 1455-1467; <http://csdl.computer.org/dl/trans/tp/2002/11/i1455.pdf>