



# Geração automática de código LaTeX a partir de expressões matemáticas manuscritas

Acadêmico: Emanuel H. Farias

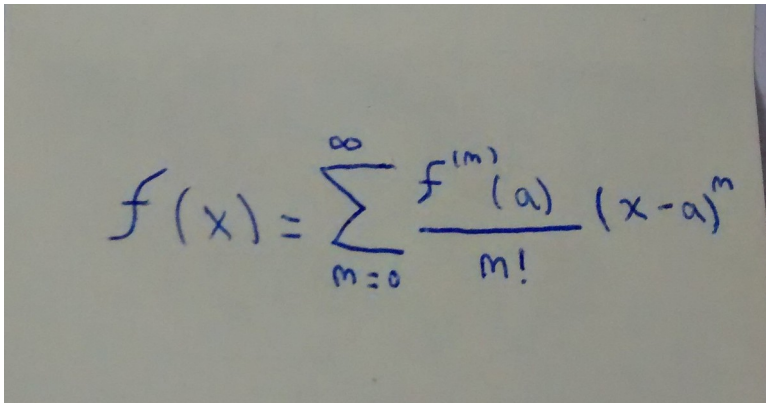
Orientador: Dr. André Tavares da Silva

# Roteiro

- Objetivo
- Etapas do reconhecimento de expressões
- Escopo do TCC
- Resultados
- Conclusão
- Trabalhos Futuros

# Objetivo

- Reconhecedor de expressões matemáticas manuscritas
- Aplicação web
  - *Upload* de uma imagem, ou foto via *webcam*


$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

Entrada



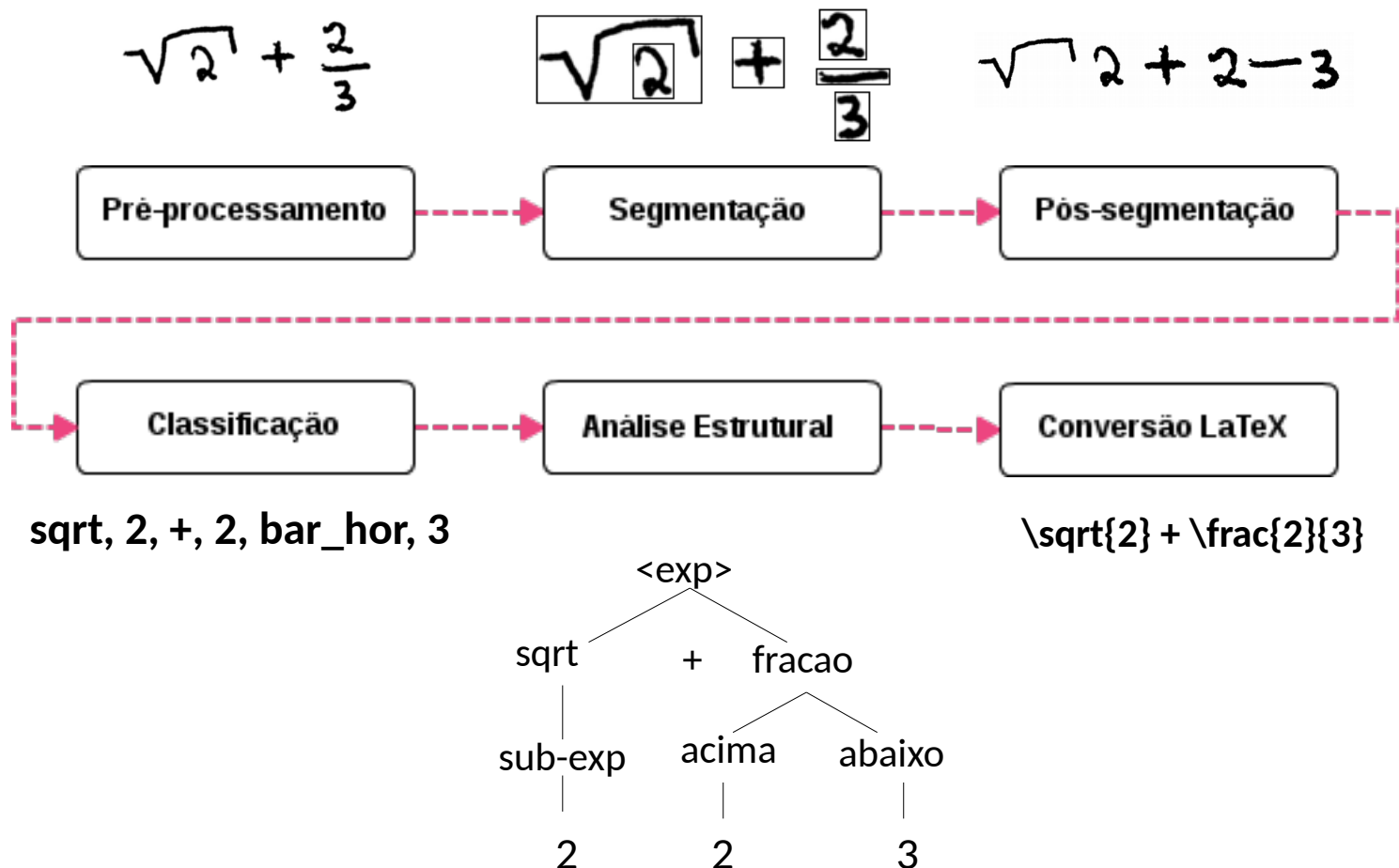
$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

LaTeX:

`f(x) = \sum_{n=0}^{\infty}`  
`\frac{f^{(n)}(a)}{n!}(x-a)^n`

Saída

# Etapas do Reconhecimento



# Pré-processamento

- Conversão da imagem para **tons de cinza**
- **Redução de ruídos** com filtro de mediana 7x7
- **Limiarização Adaptativa**

$$f(x) = \sum_{m=0}^{\infty} \frac{f^{(m)}(a)}{m!} (x-a)^m$$

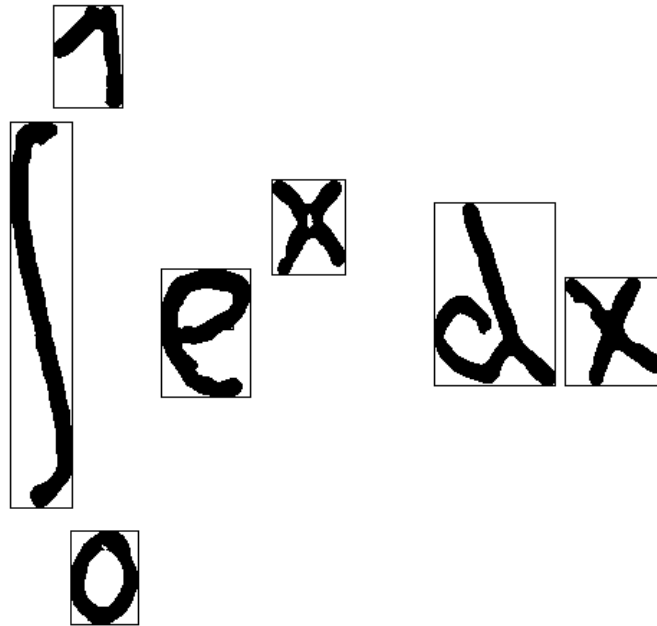


$$f(x) = \sum_{m=0}^{\infty} \frac{f^{(m)}(a)}{m!} (x-a)^m$$

# Segmentação

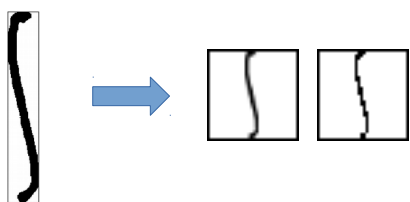
**Isolar** cada símbolo da expressão matemática

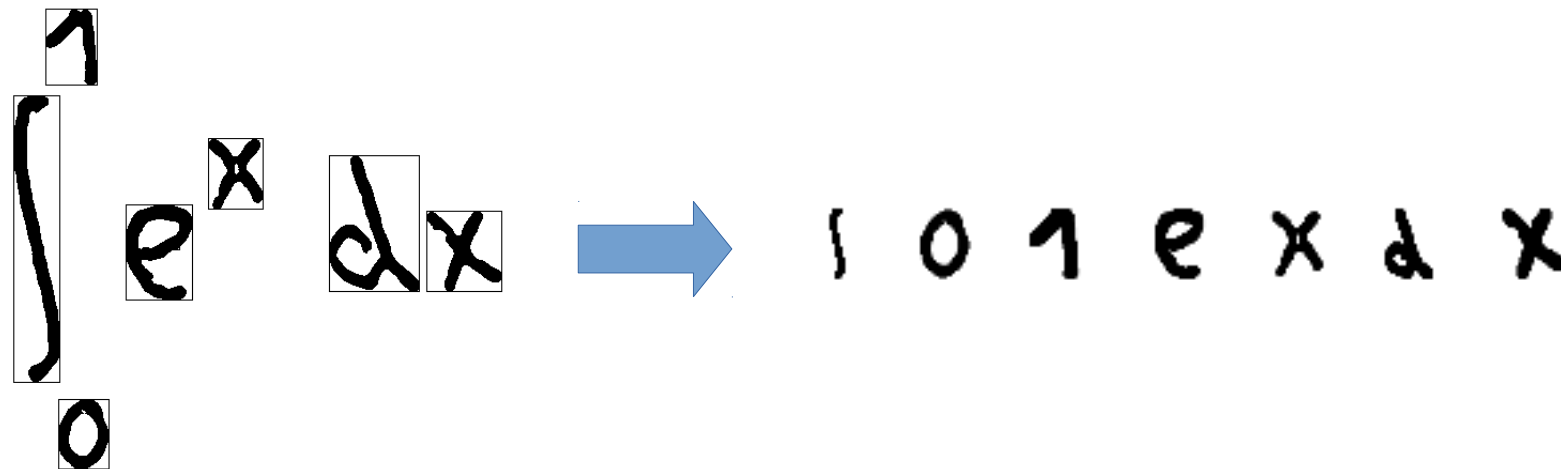
Método: Componentes Conexos




# Pós-segmentação

Tratar os símbolos antes de serem classificados.

- Dilatar 
- Redimensionar para 32 x 32



# Reconhecimento/Classificação

  
integral 0 1 e x d x

- Atribuir rótulos as imagens
- Aprendizado de Máquina Supervisionado
- Classificador: Redes Neurais Convolucionais (CNN)



# Treinamento

- 4 **datasets** foram utilizadas:
  - HWRT  
Símbolos matemáticos variados, dígitos, alfabeto latino
  - Chars74K  
Dígitos e alfabeto latino
  - MNIST  
Dígitos (0-9)
  - Base do próprio autor  
Símbolos matemáticos variados

# Treinamento

- O conjunto de dados final possui **27 classes**
  - **Dígitos:** 0-9
  - **Caracteres:** a, b, c, d, e, f, x, y, z
  - **Delimitadores:** (, )
  - **Operadores:** +, -, integral, somatório e raiz
  - **Outros:** ponto
- A *dataset* final possui 66271 símbolos divididos em 3 conjuntos:
  - Treino (60%)
  - Validação (20%)
  - Teste (20%)

# Arquitetura da CNN

- Adaptação da arquitetura da LeNet-5 (LeCun, 1998)

---

**Entrada:** imagem  $32 \times 32$  normalizada em  $[0.0, 1.0]$

---

**camada 1:** 32 *features maps* de convolução  $5 \times 5$  com *stride* = 1

---

Ativação com Rectified Linear Unit (ReLU)

---

**camada 2:** *Pool* ( $2 \times 2$ )

---

**camada 3:** 15 *features maps* de convolução  $3 \times 3$  com *stride* = 1

---

Ativação ReLU

---

**camada 4:** *Pool* ( $2 \times 2$ )

---

*Dropout* de 20%

---

**camada 5:** Conversão de matriz para vetor

---

**camada 6:** 256 neurônios totalmente conectados com a camada anterior

---

Ativação ReLU

---

**camada 7:** 128 neurônios totalmente conectados com a camada anterior

---

Ativação ReLU

---

**camada 8:** 27 neurônios totalmente conectados com a camada anterior

---

Ativação *softmax*

---

**Saída:** vetor com 27 elementos variando de  $[0.0, 1.0]$

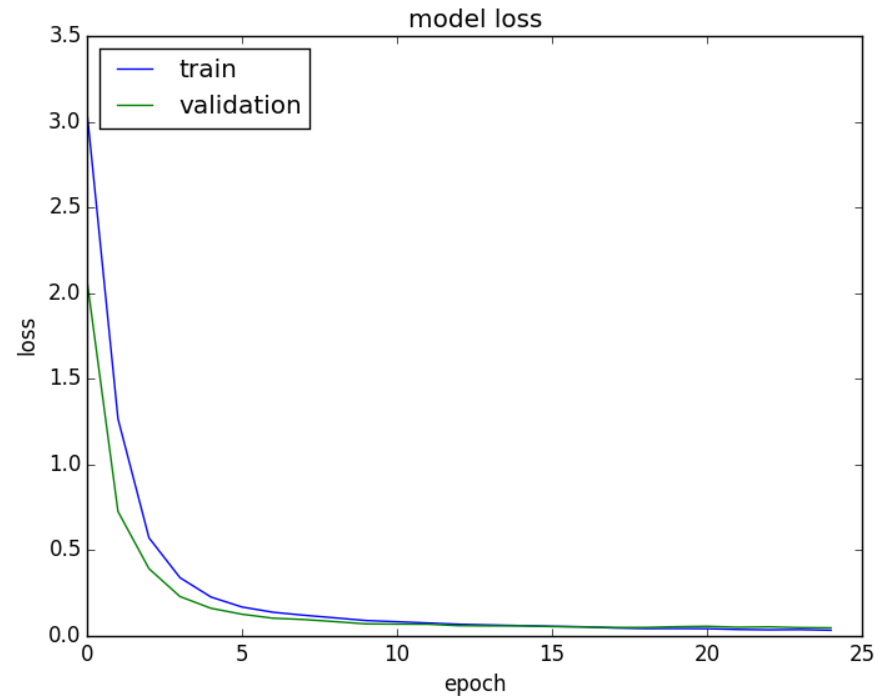
---

# Parâmetros Treinamento

- **Função de erro:** *cross-entropy* ou *log loss*
- **Otimizador:** *ADAM* (*gradiente descendente estocástico*)
- **Inicialização dos pesos:** *aleatoriamente seguindo a distribuição gaussiana.*
- **Iterações:** 25
- **Batch-size:** 500

# Resultados do Treinamento

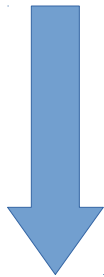
- Erro: 1.43% (em relação ao conjunto de testes)



# Criação da Árvore de expressões matemáticas

Imagem Original

$(2 + 3) * 4$



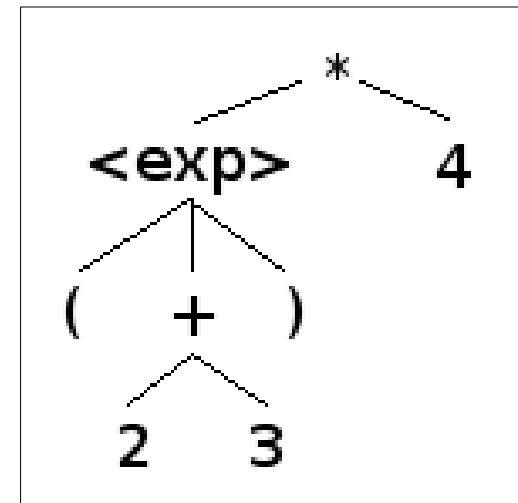
Classificação

(, 2, +, 3, ), \*, 4

Símbolos já identificados



Árvore de Expressões

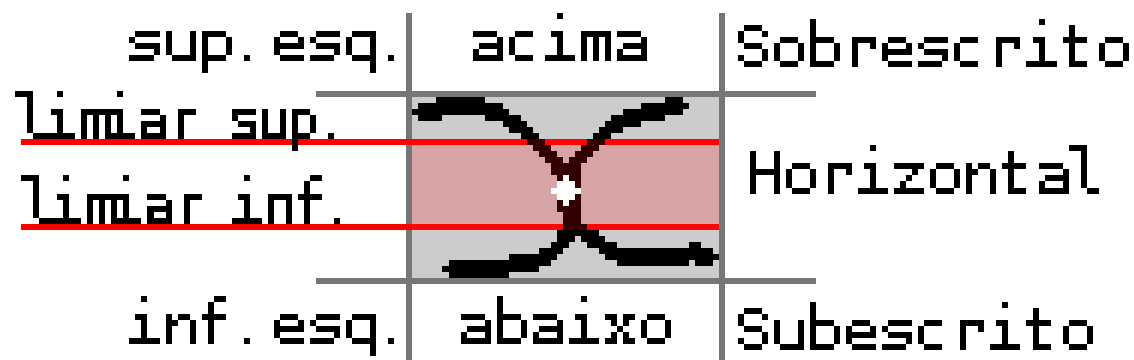


# Criação da Árvore de expressões matemáticas

- **Análise Estrutural**

- Minimum Spanning Tree (MATSAKIS, **1999**)
- Baseline Structure Tree (ZANIBBI; BLOSTEIN; CORDY, **2001**)
- BST e MST (TAPIA, **2005**)

# Dominância e Aspecto Comportamental

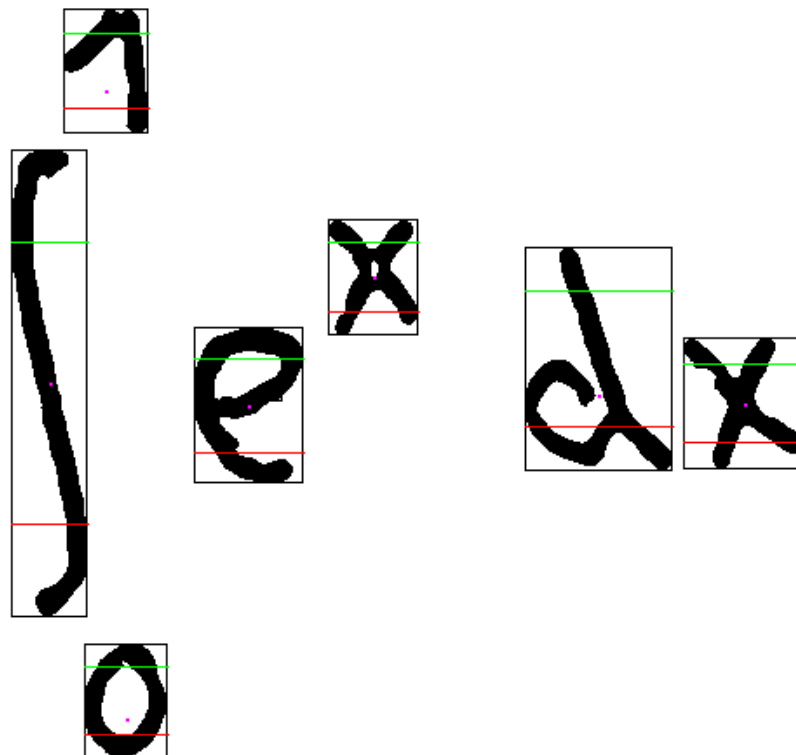




# Centroides e Limiares

## Grafia:

- Central
- Ascendente
- Descendente

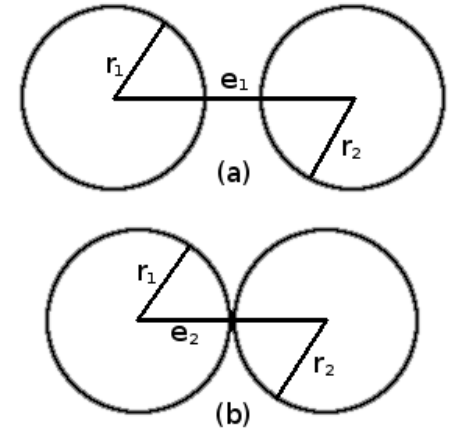


# Grafo

- Construção de um grafo totalmente conectado (todos os símbolos conectam com todos)

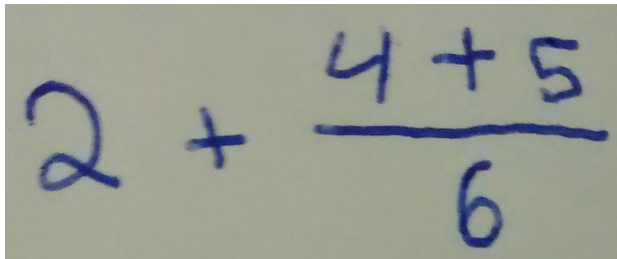
## Pesos das arestas:

- Se um símbolo domina o outro, o peso da aresta é a menor distância (aproximada) entre os bounding-boxes dos símbolos
- Caso contrário, o peso é a distância euclidiana entre os centroides dos símbolos



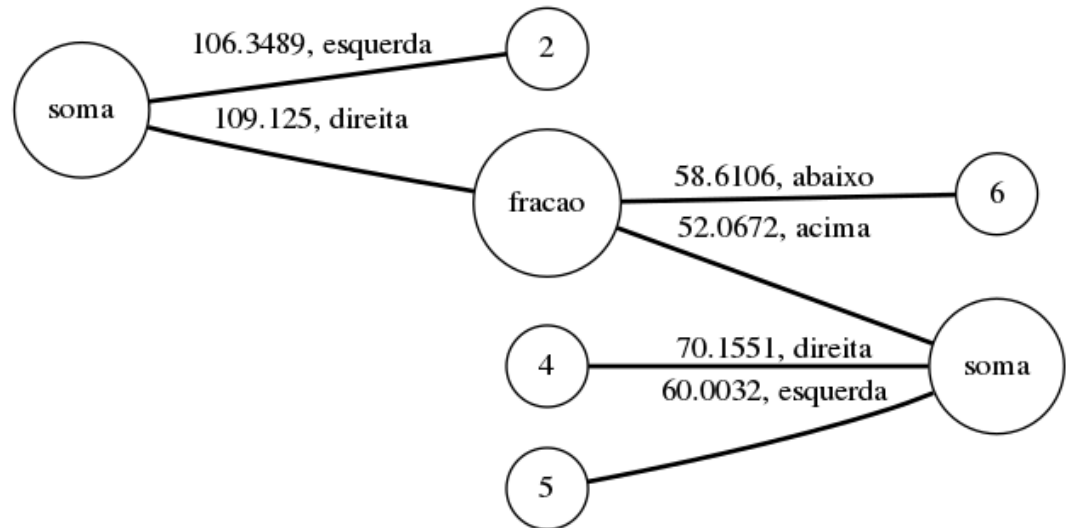
# MST

- Aplicar Prim ou Kruskal no grafo



Handwritten calculation showing the sum of 2 and the fraction 4/5, resulting in 14/5.

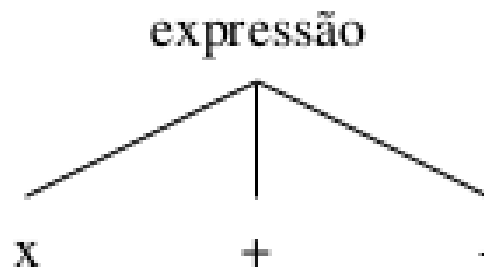
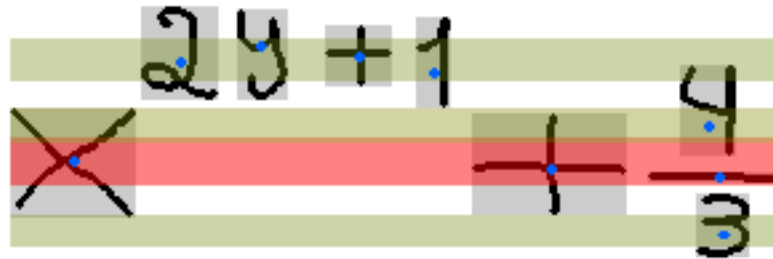
$$2 + \frac{4}{5} = \frac{14}{5}$$



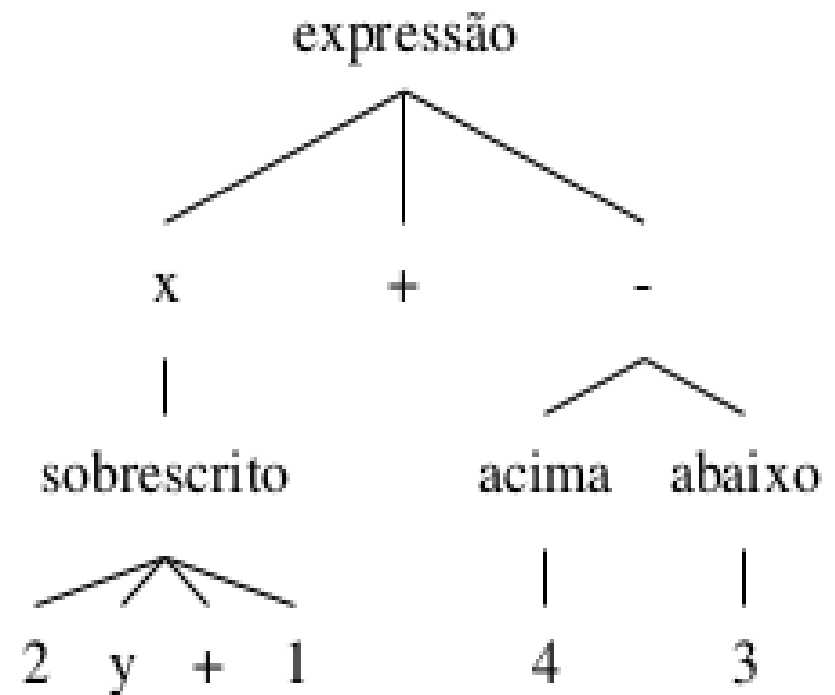
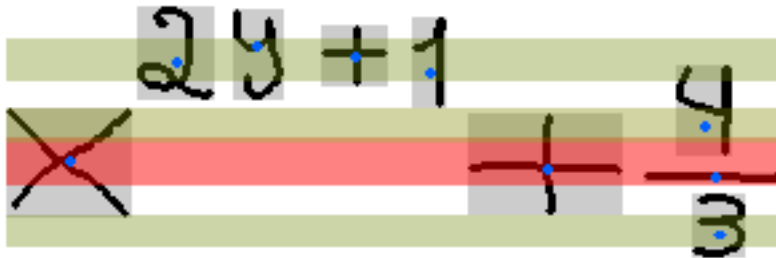
# BST

- Construção da *Árvore de Baselines* se divide em três passos:
  - 1) Encontrar a ***baseline*** principal
  - 2) Encontrar as **demais baselines** que estão associadas com a principal (analisando a dominância e MST)
  - 3) Aplicar o algoritmo **recursivamente** para cada nova *baseline encontrada*

# Análise Estrutural: Árvore de *Baselines*



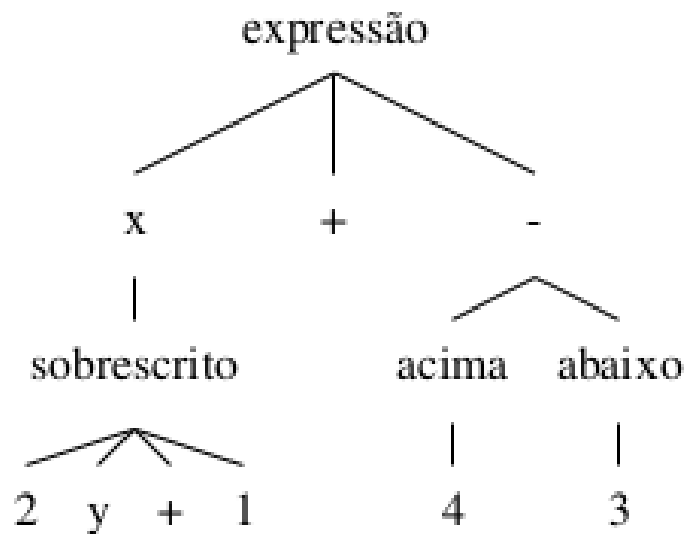
# Análise Estrutural: Árvore de *Baselines*



# Análise Estrutural: resumo

- 1) Definir centroides e limiares para cada símbolo conforme a grafia
- 2) Construir grafo e MST da expressão matemática
- 3) Construir árvore de *baselines* (BST)

# Conversão da Árvore para código Latex



Estrutura de Dados



Código LaTeX:

**`x^{2y+1} + \frac{4}{3}`**



# Escopo

- Todos os símbolos devem estar separados



- Expressões que podem ser reconhecidas:
  - Aritméticas básicas (soma, subtração, multiplicação e divisão)
  - Polinômios
  - Funções Racionais (divisão de polinômios)
  - Funções Exponenciais
  - Integrais e Somatórios

# Escopo

- A região da imagem selecionada (crop) para o reconhecimento deve ter **apenas uma expressão** matemática
- A escrita da expressão deve ser em **folhas que não possuam linhas**, ou seja, folhas de caderno não são aceitas
- A expressão deve ser escrita em uma **única linha**, ou seja, não pode haver quebra da expressão e continuar na próxima linha
- A expressão deve ser escrita de forma **alinhada na horizontal**



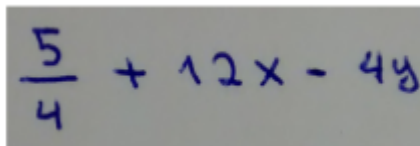
A handwritten mathematical expression  $3x - y + 4$  is shown on a tilted background, illustrating a violation of the horizontal alignment rule.

# Escopo: outras restrições

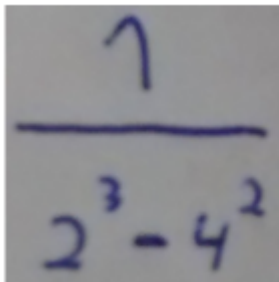
- Expressões matemáticas que o sistema não reconhece:
  - Funções trigonométricas
  - Números com casas decimais (vírgulas e pontos)
  - Raizes (algumas reconhecem)
- Porém, com ajustes na *dataset* e e na etapa de análise estrutural, estas expressões podem ser reconhecidas

# Resultados

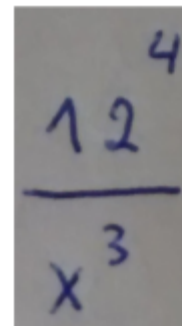
- Foram coletadas expressões matemáticas de seis pessoas
- A seguir algumas expressões que foram reconhecidas **corretamente**:


$$\frac{5}{4} + 12x - 4y$$

Resultado:  $\frac{5}{4} + 12x - 4y$

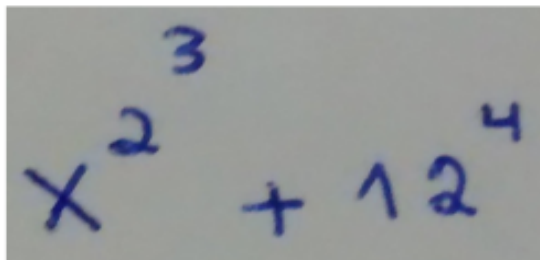

$$\frac{1}{2^3 - 4^2}$$

Resultado:  $\frac{1}{2^3 - 4^2}$


$$\frac{12^4}{x^3}$$

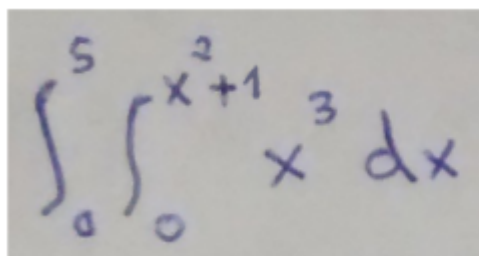
Resultado:  $\frac{12^4}{x^3}$

# Resultados: corretas



A photograph of a handwritten mathematical expression in blue ink on a light-colored surface. The expression is  $x^{2^3} + 12^4$ . The '2' in the first term has a '3' as a superscript, and the '12' in the second term has a '4' as a superscript.

Resultado:  $x^{2^3} + 12^4$

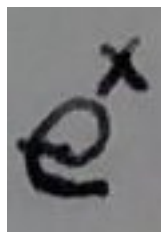


A photograph of a handwritten mathematical expression in blue ink on a light-colored surface. The expression is  $\int_0^5 \int_0^{x^2+1} x^3 dx$ . The outer integral is from 0 to 5, and the inner integral is from 0 to  $x^2+1$ .

Resultado:  $\int_0^5 \int_0^{x^2+1} x^3 dx$

# Resultados: incorretas

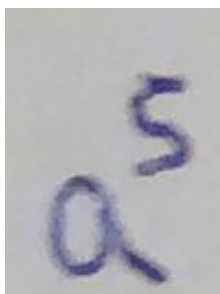
Erros se propagam nas etapas seguintes



200  
↓  
4

100  
↓  
1a

# Resultados: incorretas



A photograph of the handwritten expression  $x + \sum_{x=1}^{123456}$  on a piece of paper. The image is used to illustrate incorrect recognition results, as the summation symbol and its limits are not properly identified.

# Conclusão

- Este é um trabalho completo (implementa todas as etapas) de um reconhecedor *offline* de expressões matemáticas manuscritas.
- O sistema apresentou bons resultados, mesmo utilizando métodos comuns da área de processamento de imagens
- Para implementar um sistema mais robusto, todas as etapas podem ser melhoras com métodos mais avançados.



# Trabalhos Futuros

- **Pré-processamento:** estratégias mais avançadas para tratar documentos utilizam *clustering* para separar o fundo da escrita.
- **Segmentação:** utilizar o método *Projection Profile Cutting* que utiliza histogramas verticais e horizontais, ou ainda, utilizar redes neurais próprias para auxiliar a segmentação.
- **Treinamento:** aumentar a *dataset* inserindo o alfabeto latino completo, alfabeto grego, e outros símbolos matemáticos ( $>$ ,  $<$ ,  $\{$ ,  $\}$ ,  $[$ ,  $]$ )
- **Análise Estrutural:** definir limiares e classes comportamentais específicas para cada símbolo.
- Adicionar etapa de **correção de erros** pós análise estrutural, que analisa a semântica dos símbolos (ex: abre e fecha parenteses)



# Geração automática de código LaTeX a partir de expressões matemáticas manuscritas

Acadêmico: Emanuel H. Farias

Orientador: Dr. André Tavares da Silva