

EXPRESSMATH: ANÁLISE ESTRUTURAL DE EXPRESSÕES MATEMÁTICAS MANUSCRITAS



Bruno Yoiti Ozahata, Ricardo Sider e Nina S. T. Hirata (orientadora)



Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

Conselho Nacional de Desenvolvimento Científico e Tecnológico

Palavras-Chave: Expressão matemática; Escrita manuscrita; Análise estrutural; Baseline; Árvore geradora mínima.

Resumo

Apesar do reconhecimento de escrita ser um tema de pesquisa já bastante explorado, ainda existem muitos desafios. Um desses desafios é o reconhecimento de notação matemática, um caso em que a quantidade de possíveis símbolos é grande e sua disposição espacial complexa. O reconhecimento de expressões matemáticas é geralmente dividido em duas etapas: (1) segmentação e reconhecimento dos símbolos, e (2) análise estrutural. Neste trabalho apresentamos uma abordagem encontrada na literatura recente para análise estrutural de expressões matemáticas manuscritas e sua implementação. A abordagem explora conceitos como *baselines* e *árvores geradoras mínimas* para determinar a estrutura de uma expressão.

1. Regiões e dominância entre símbolos matemáticos

UMA expressão matemática consiste de um arranjo espacial de diversos símbolos no plano bidimensional, na qual a posição relativa entre os símbolos é determinante da relação entre eles e, consequentemente de suas funções do ponto de vista semântico na expressão.

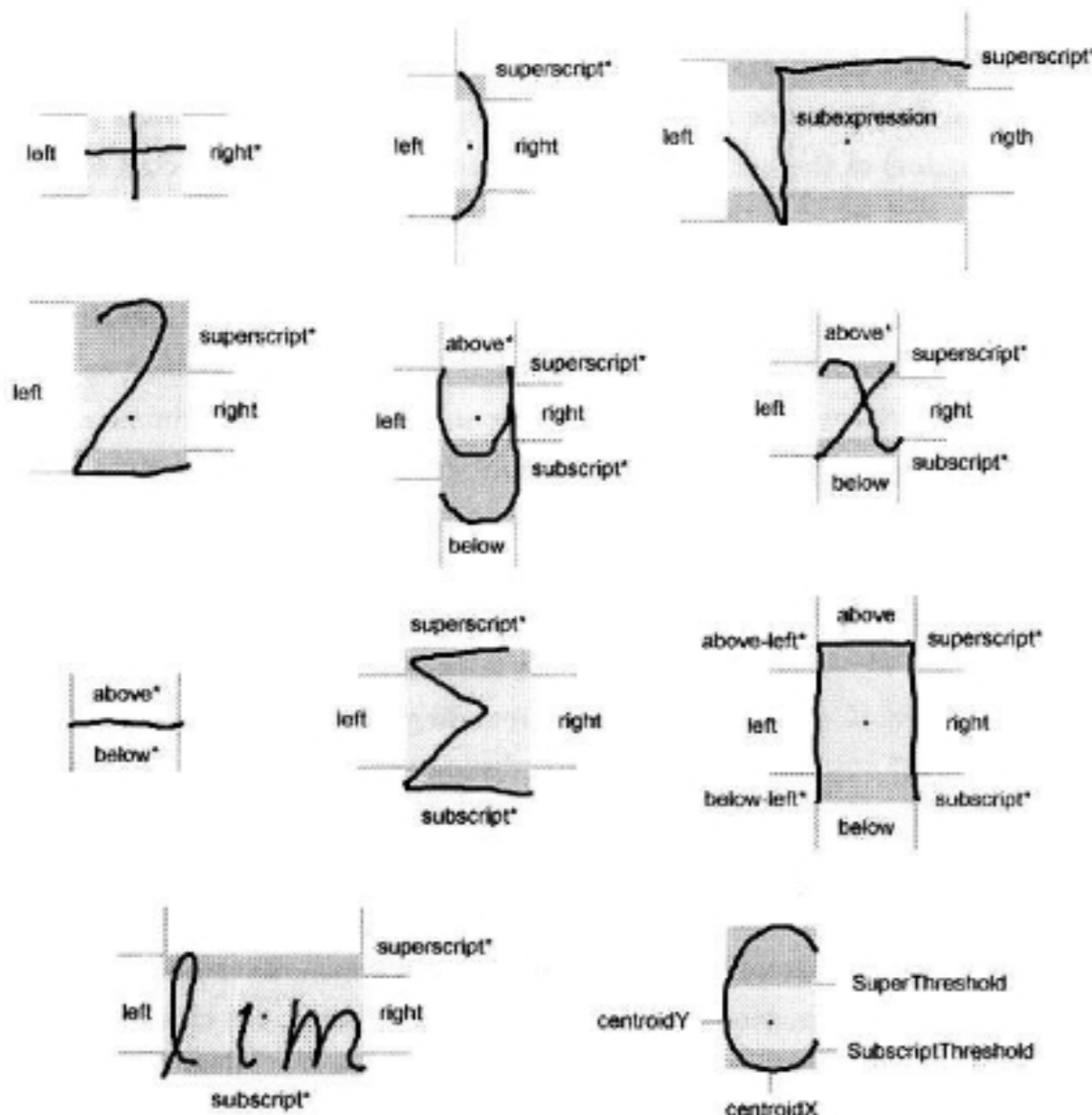


Figura 1: Alguns símbolos com seus atributos e regiões (Fonte: [4])

As relações de dominância numa expressão matemática são determinadas implicitamente ou explicitamente pela posição e tamanho relativo dos símbolos na expressão. Definimos dominância em função da área onde esperamos encontrar atributos de um símbolo (abrangência) do seguinte modo: um símbolo *a* domina sobre outro símbolo *b*, se *b* está na área de abrangência de *a*, e *a* não está na área de abrangência de *b*. Em casos em que a dominância não pode ser determinada dessa forma, consideramos ainda o tamanho relativo dos símbolos, e a ordem de leitura da esquerda para a direita.

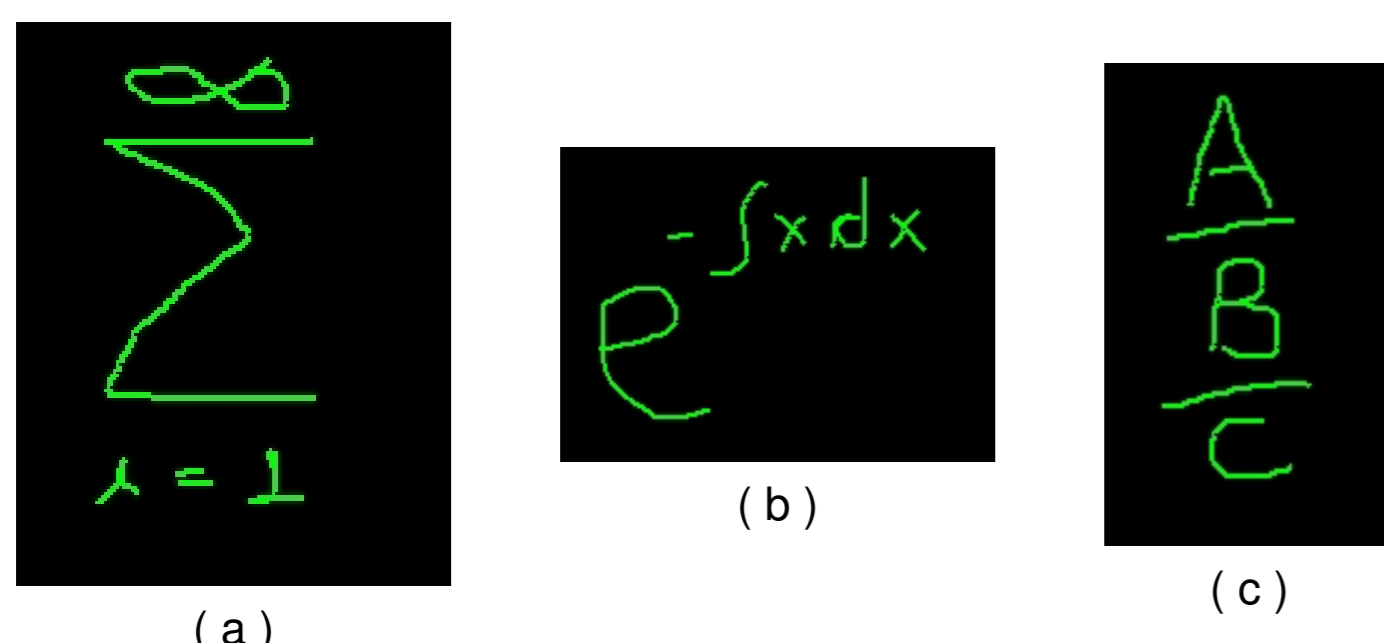


Figura 2: Expressões onde (a) Dominância é determinada pela abrangência. (b) Dominância é determinada pelo tamanho. (c) É difícil determinar a dominância dos operadores

2. Baselines

Uma *baseline* é uma lista ordenada de símbolos, que representa um arranjo horizontal na expressão, de modo que podemos representar uma fórmula matemática como um conjunto de *baselines* aninhadas hierarquicamente.

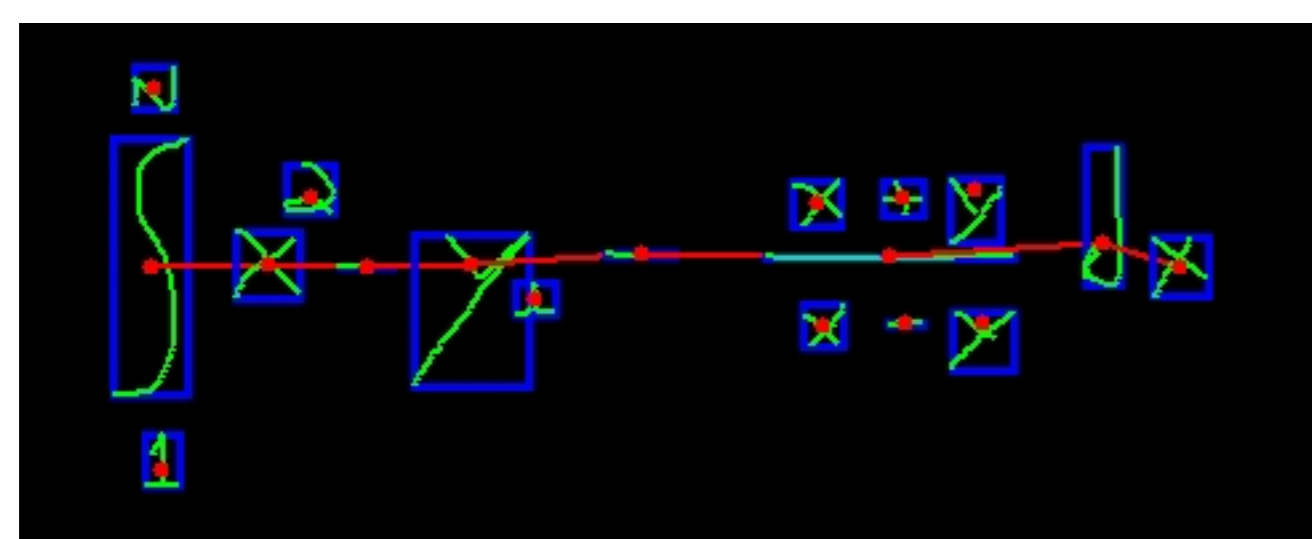


Figura 3: Uma expressão simples e sua *baseline* principal.

Na expressão da figura, tanto o numerador quanto o denominador da barra de divisão são *baselines*, que satisfazem a relação "acima", e "abaixo" com a *baseline* principal, respectivamente.

3. Algoritmo para análise estrutural

A partir de uma lista de símbolos previamente reconhecida pelo classificador, com informações básicas sobre cada símbolo, dá-se início ao processo de análise estrutural, que segue o seguinte fluxo:

1. Determinar *baseline* principal;
2. Calcular a *árvore geradora mínima*;
3. Construir listas com os filhos dos símbolos na *baseline* principal, rotuladas com a relação espacial que satisfazem com ele;
4. Aplicar a recursão para cada uma das listas não vazias.

Construímos a *baseline* principal buscando símbolos dominantes na expressão que sejam adjacentes. A chave para um reconhecimento satisfatório é uma definição robusta de adjacência.

Após o reconhecimento da *baseline* principal, associamos cada símbolo a um vértice num grafo totalmente conexo, e construímos sua *árvore geradora mínima* utilizando o algoritmo de Prim a partir da *baseline* principal. As arestas da MST são representantes das relações mais relevantes na expressão. Para o cálculo dos pesos das arestas, o mais importante nessa abordagem, leva-se em consideração o tipo dos símbolos envolvidos, a sua participação numa *baseline* pré-reconhecida, dominância, e distância.

Símbolos com limites e barras de divisão requerem atenção especial, pois frequentemente seus limites são escritos longe dos seus operadores. Para corrigir esse problema, multiplicamos o peso dessas arestas por um fator $0 < \alpha < 1$. A figura 4 mostra um exemplo de reconhecimento com e sem esse fator.

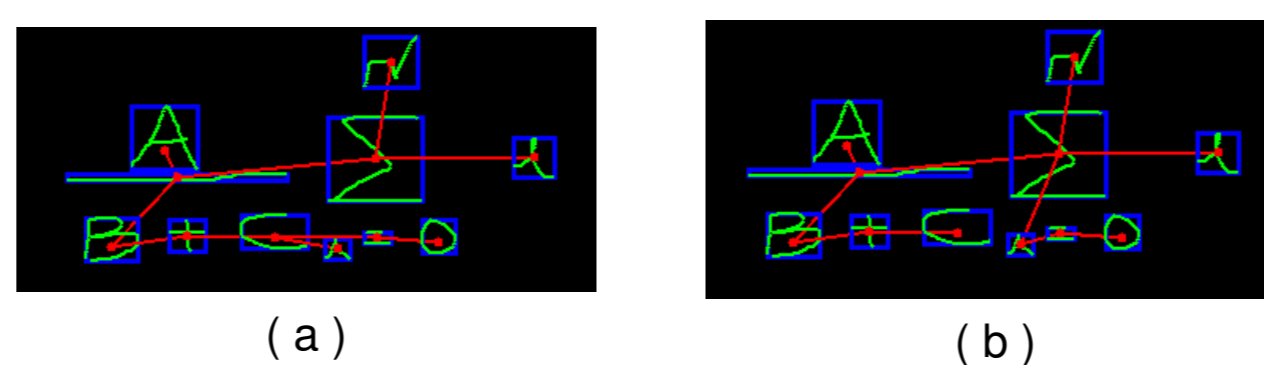


Figura 4: *Árvore geradora mínima* da expressão (a) Sem o fator α . (b) Usando o fator α .

Após a geração da MST da expressão, criam-se listas de símbolos a partir de cada vértice filho de um símbolo na *baseline* principal, identificadas pela relação espacial que satisfazem para com o seu símbolo pai, e aplicamos o algoritmo recursivamente para cada uma delas.

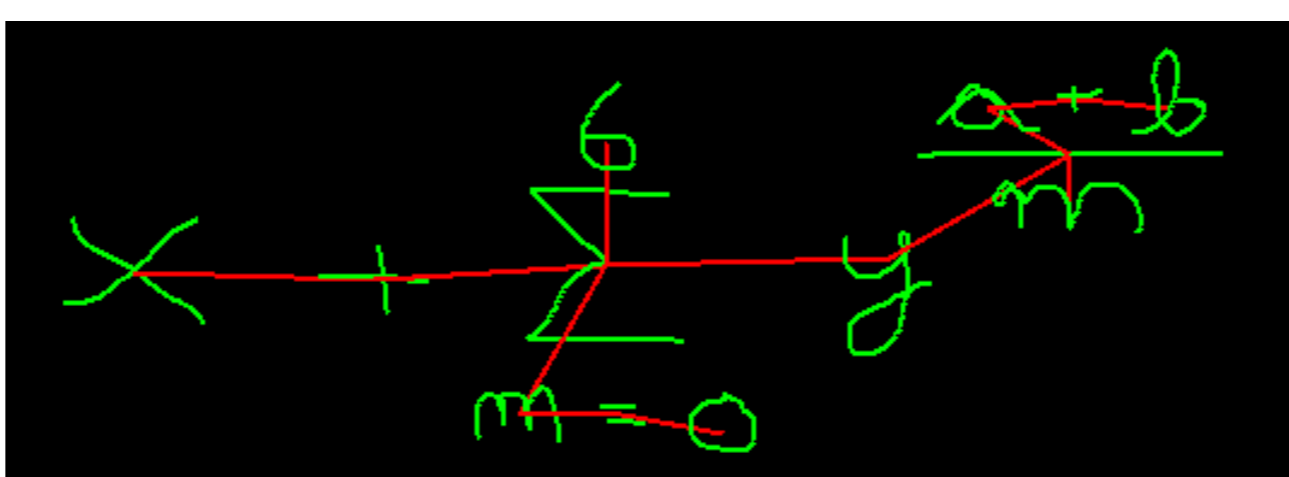


Figura 5: *Árvore geradora mínima final* de uma expressão

4. Interface com o usuário e integração com o reconhecedor

A interface do Express-Math foi criada a partir de alterações na interface do Math-Picasso, feitas em conjunto com o grupo que está trabalhando num novo algoritmo de reconhecimento.

Em particular, disponibilizamos uma interface para classificação manual dos símbolos, para garantir que a análise estrutural pudesse ser realizada a partir de todos os símbolos identificados corretamente. É muito importante para o nosso método que os caracteres estejam reconhecidos de maneira correta.

A seguir apresentamos telas do nosso sistema, em diferentes estágios do reconhecimento.

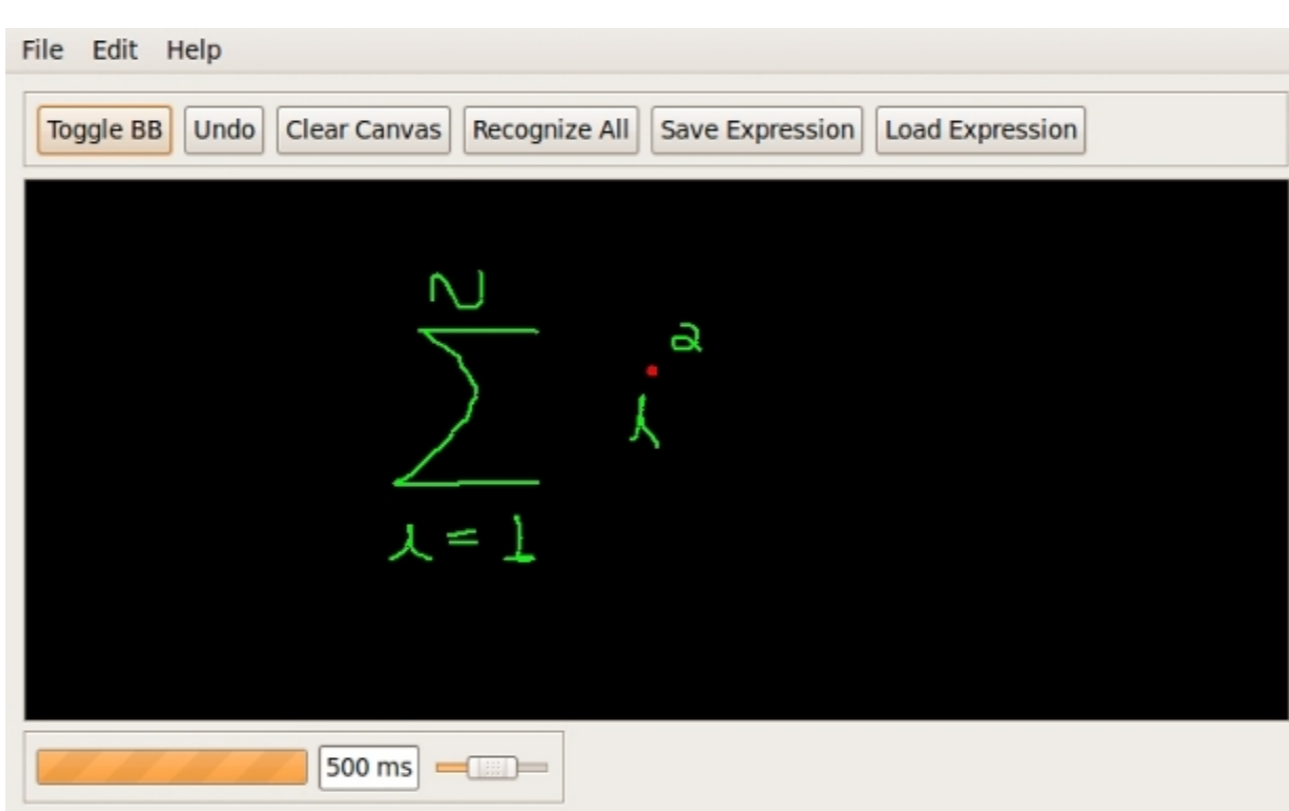


Figura 6: Expressão matemática a ser reconhecida

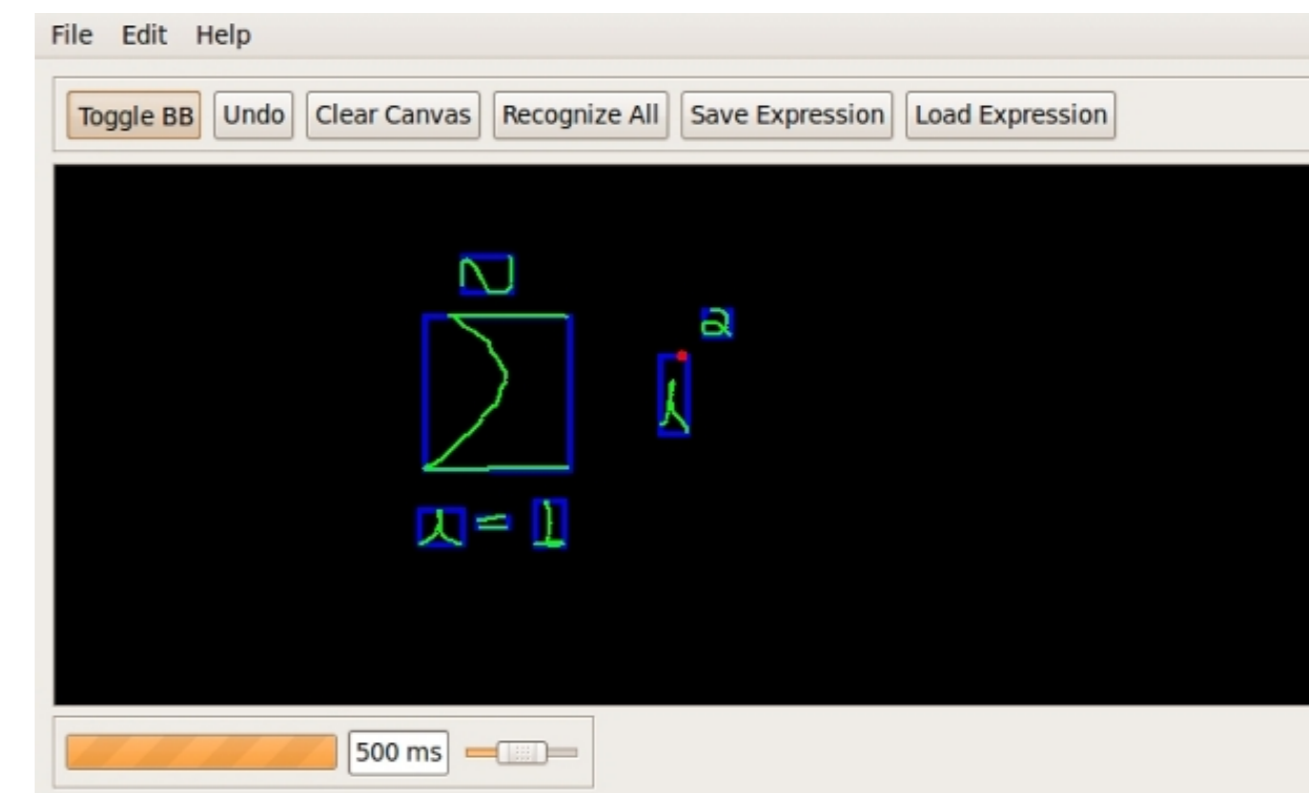


Figura 7: Expressão segmentada com retângulos envoltórios

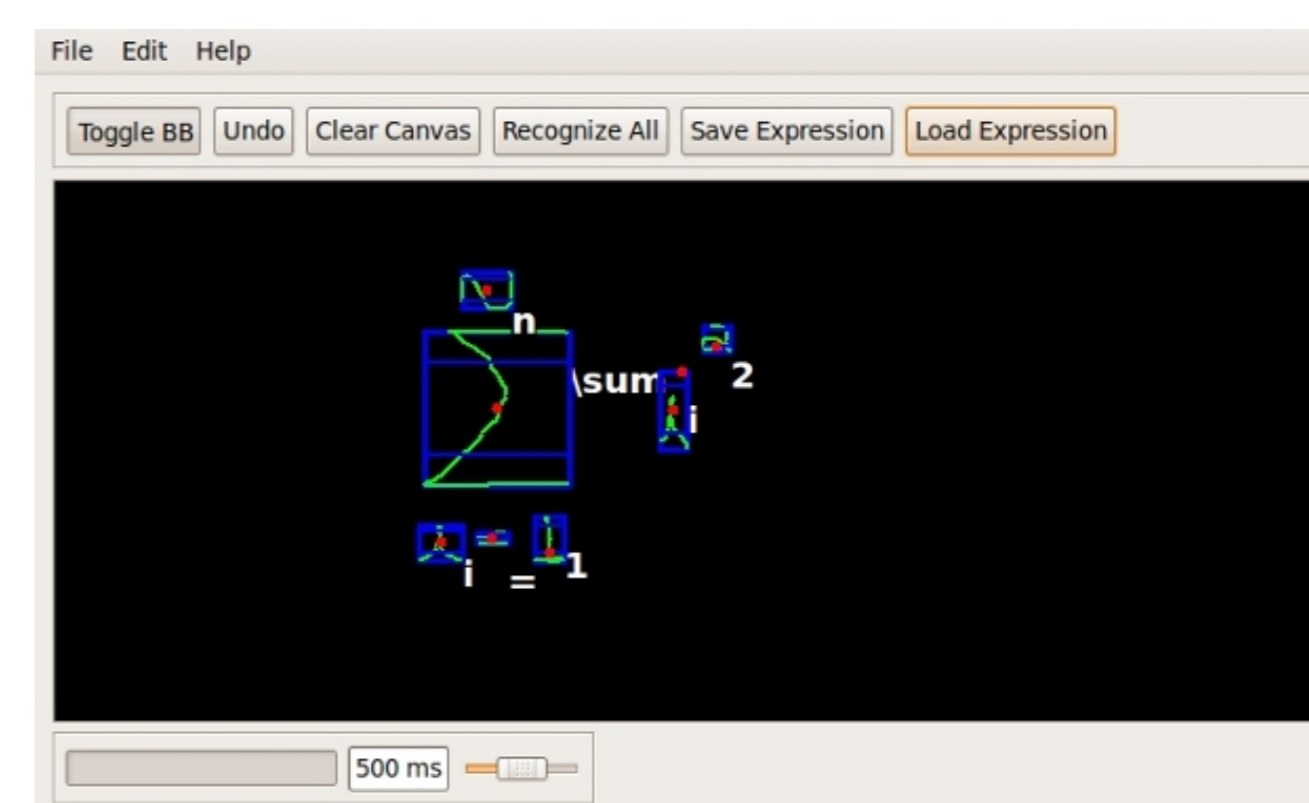


Figura 8: Expressão reconhecida: símbolos com etiqueta, centróide e limites

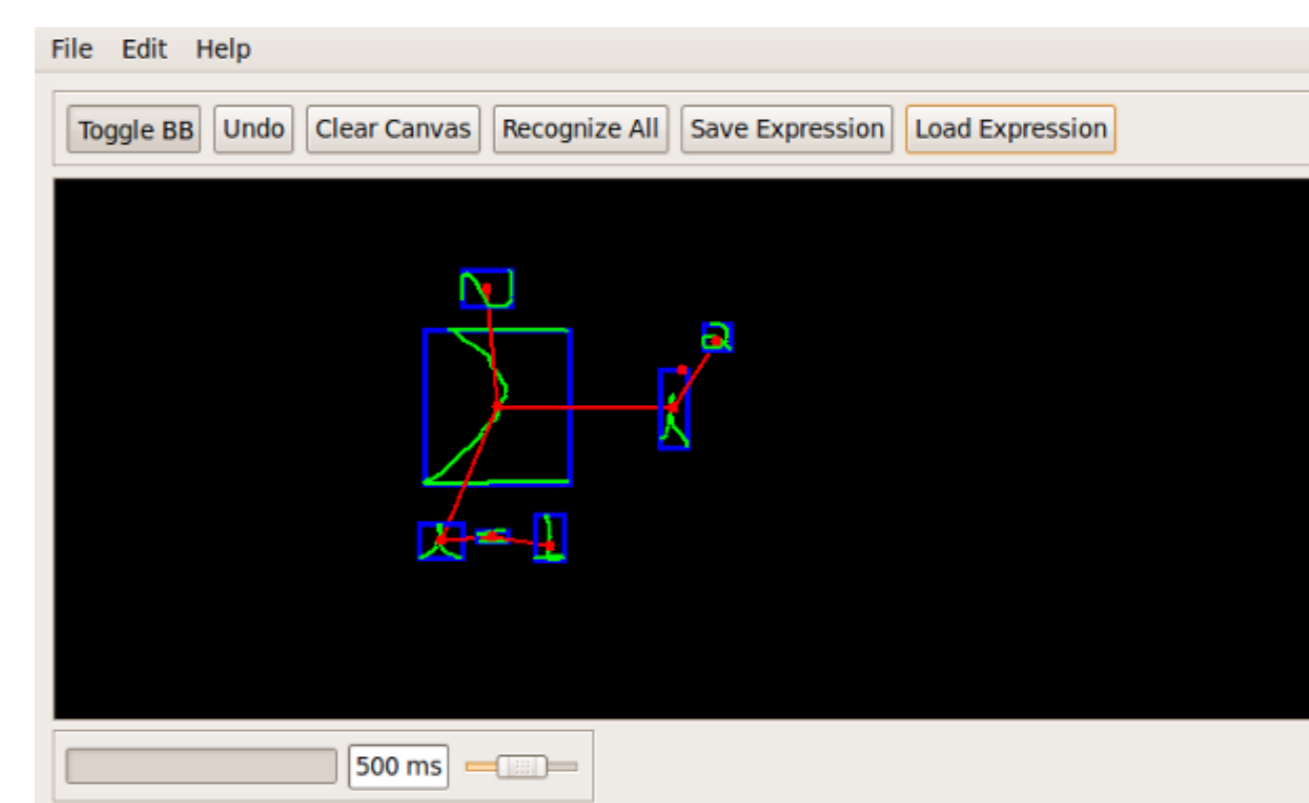


Figura 9: *Árvore geradora mínima final* da expressão

5. Resultado Final

Após a análise estrutural a *árvore geradora mínima final* da expressão é percorrida, e por meio de um algoritmo simples geramos a representação da expressão original em \LaTeX . Essa linguagem foi escolhida por sua ampla aceitação na produção de textos científicos no meio acadêmico, a abertura de seu código e especificações, ampla gama de símbolos reconhecidos.

É importante ressaltar que a partir da *árvore geradora mínima* da expressão é possível reconstruir a fórmula original em praticamente qualquer formato digital estruturado por meio de uma função recursiva simples.

6. Conclusão

Descrevemos um método conhecido e recente para análise estrutural de expressões matemáticas manuscritas fortemente baseado no conceito de dominância entre símbolos e operadores e que faz uso de *baselines* e *árvores geradoras mínimas*.

Descrevemos também um sistema computacional que implementamos para a análise estrutural de expressões matemáticas manuscritas, baseado no método descrito. Nossa implementação gera ao final da análise estrutural a representação da expressão em \LaTeX um dos formatos mais utilizados na criação e edição de notação matemática.

Segundo os autores dessa abordagem, ela ainda apresenta deficiências no que diz respeito à eficiência, robustez e reconhecimento de estruturas complexas como, por exemplo, matrizes e sistemas de equações. Essas mesmas deficiências foram verificadas nesse projeto. Algumas expressões mais complexas como as que incluem estruturas tabulares, como matrizes, definições de funções e operadores com limites em pilha, não são corretamente reconhecidas.

Referências

- [1] Ana Paula Santos de Mello, Eduardo Yutaca Komatsu, Fábio Marcos Eiji Okuda, and Leonardo Ka Wah Hing, *Math picasso - segmentação e reconhecimento de caracteres em expressões matemáticas manuscritas*, 2007, <http://www.linux.ime.usp.br/~eijl/mac499/index.php>.
- [2] Nina S. T. Hirata, *ExpressMath - reconhecimento de expressões matemáticas*, <http://www.vision.ime.usp.br/~nina/projetos/expressmath/>, 2008.
- [3] Nicholas E. Matsakis, *Recognition of handwritten mathematical expressions*, Master's thesis, Massachusetts Institute of Technology, 1999.
- [4] E. Tapia and R. Rojas, *Recognition of on-line handwritten mathematical expressions using a minimal spanning tree construction and symbol dominance*, 2004.
- [5] R. Zanibbi, D. Blostein, and J. R. Cordy, *Recognizing mathematical expressions using tree transformation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.