**Rotulação de símbolos matemáticos manuscritos
via casamento de expressões**

Willian Yukio Honda

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Ciências da Computação
Orientadora: Prof^a. Dr^a. Nina S. T. Hirata

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

São Paulo, fevereiro de 2013

Rotulação de símbolos matemáticos manuscritos via casamento de expressões

Esta dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa realizada por Willian Yukio Honda em 23/01/2013.

O original encontra-se disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof^a. Dr^a. Nina Sumiko Tomita Hirata (orientadora) - IME-USP
- Prof. Dr. Carlos Eduardo Ferreira - IME-USP
- Prof. Dr. Alexandre Noma - UFABC

Agradecimentos

Primeiramente, gostaria de agradecer a Deus por todas as conquistas em minha vida.

À minha querida família pelos valores, princípios, incentivo, apoio, força e encorajamento. Sem eles, não seria nada do que sou hoje.

À Roberta Sayuri da Costa Iraha, por todo o carinho e mensagem positiva que resultaram em motivação.

À Prof^a. Dr^a. Nina S. T. Hirata pelos ensinamentos, auxílio e paciência, elementos essenciais para meu aprendizado. Muito obrigado pela excelente orientação e condução em todo o meu mestrado.

Também aproveito para agradecer a todos os mestres que guiaram minha vida e me auxiliaram a construir a base do meu conhecimento.

Ao Alexandre Noma, pela inspiração e importante contribuição para os alicerces deste trabalho. Sem eles, o mesmo não teria toda força para seu sucesso.

Ao Frank Dennis Julca Aguilar pela excelente ferramenta visual do ExpressMatch e por toda a ajuda e contribuição no desenvolvimento do trabalho, essenciais para a realização dos experimentos, análise dos resultados e qualidade da apresentação.

À todos os meus amigos que diretamente ou indiretamente me incentivaram a lutar por essa conquista.

Muito obrigado.

Resumo

O problema de reconhecimento de expressões matemáticas manuscritas envolve três subproblemas importantes: segmentação de símbolos, reconhecimento de símbolos e análise estrutural de expressões. Para avaliar métodos e técnicas de reconhecimento, eles precisam ser testados sobre conjuntos de amostras representativos do domínio de aplicação. Uma das preocupações que tem sido apontadas ultimamente é a quase inexistência de base de dados pública de expressões matemáticas, o que dificulta o desenvolvimento e comparação de diferentes abordagens. Em geral, os resultados de reconhecimento apresentados na literatura restringem-se a conjuntos de dados pequenos, não disponíveis publicamente, e muitas vezes formado por dados que visam avaliar apenas alguns aspectos específicos do reconhecimento.

No caso de expressões *online*, para treinar e testar reconhecedores de símbolos, as amostras são em geral obtidas solicitando-se que as pessoas escrevam uma série de símbolos individualmente e repetidas vezes. Tal tarefa é monótona e cansativa. Uma abordagem alternativa para obter amostras de símbolos seria solicitar aos usuários a transcrição de expressões modelo previamente definidas. Dessa forma, a escrita dos símbolos seria realizada de forma natural, menos monótona, e várias amostras de símbolos poderiam ser obtidas de uma única expressão. Para evitar o trabalho de anotar manualmente cada símbolo das expressões transcritas, este trabalho propõe um método para casamento de expressões matemáticas manuscritas, no qual símbolos de uma expressão transcrita por um usuário são associados aos correspondentes símbolos (previamente identificados) da expressão modelo. O método proposto é baseado em uma formulação que reduz o problema a um problema de associação simples, no qual os custos são definidos em termos de características dos símbolos e estrutura da expressão. Resultados experimentais utilizando o método proposto mostram taxas médias de associação correta superiores a 99%.

Palavras-chave: casamento de expressões matemáticas, expressões matemáticas manuscritas, emparelhamento de grafos bipartidos, rotulação de símbolos matemáticos.

Abstract

The problem of recognizing handwritten mathematical expressions includes three important subproblems: symbol segmentation, symbol recognition, and structural analysis of expressions. In order to evaluate recognition methods and techniques, they should be tested on representative sample sets of the application domain. One of the concerns that are being repeatedly pointed recently is the almost non-existence of public representative datasets of mathematical expressions, which makes difficult the development and comparison of distinct approaches. In general, recognition results reported in the literature are restricted to small datasets, not publicly available, and often consisting of data aiming only evaluation of some specific aspects of the recognition.

In the case of online expressions, to train and test symbol recognizers, samples are in general obtained asking users to write a series of symbols individually and repeatedly. Such task is boring and tiring. An alternative approach for obtaining samples of symbols would be to ask users to transcribe previously defined model expressions. By doing so, writing would be more natural and less boring, and several symbol samples could be obtained from one transcription. To avoid the task of manually labeling the symbols of the transcribed expressions, in this work a method for handwritten expression matching, in which symbols of a transcribed expression are assigned to the corresponding ones in the model expression, is proposed. The proposed method is based on a formulation that reduces the matching problem to a linear assignment problem, where costs are defined based on symbol features and expression structure. Experimental results using the proposed method show that mean correct assignment rate superior to 99% is achieved.

Keywords: mathematical expressions matching, mathematical symbols annotation, handwritten mathematical expressions, bipartite graph matching, mathematical symbols labeling.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xiii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
1.3 Principais contribuições e resultados	3
1.4 Organização do texto	3
2 Preliminares	5
2.1 Grafos	5
2.2 Emparelhamento de grafos bipartidos balanceados	8
2.2.1 Associação de dois grafos relacionais com atributos como emparelhamento de grafo bipartido	8
2.2.2 Algoritmo Húngaro	9
2.3 Triangulação de Delaunay	12
2.4 Custo de deformação de aparência e estrutural	14
2.5 Shape context	15
3 Reconhecimento de expressões matemáticas manuscritas	19
3.1 Escrita online	19
3.2 Panorama sobre o reconhecimento de expressões matemáticas manuscritas	20
3.2.1 Segmentação	21
3.2.2 Reconhecimento de símbolos	22
3.2.3 Análise estrutural	22
3.2.4 Avaliação	24
3.3 Abordagem proposta	27
3.4 Trabalhos relacionados	29

4	Casamento de expressões matemáticas	33
4.1	Formulação do problema de associação	33
4.2	Representação de expressões matemáticas por grafos	34
4.3	Normalização espacial e de escala	35
4.4	Custos de associação	36
4.5	Avaliação das associações	39
5	Resultados Experimentais	41
5.1	Descrição dos conjuntos de dados	41
5.1.1	Conjunto A	41
5.1.2	Conjunto B	43
5.2	Avaliação de parâmetros	45
5.3	Validação	64
6	Conclusões	65
6.1	Trabalhos Futuros	65
A	Expressões do conjunto de dados B	67
	Referências Bibliográficas	73

Lista de Figuras

2.1	Exemplos de grafos completos.	6
2.2	Exemplo de um grafo bipartido.	6
2.3	Um grafo bipartido completo com três vértices em cada parte.	7
2.4	Diferenças entre um emparelhamento maximal e emparelhamento máximo em um mesmo grafo [9].	7
2.5	Emparelhamento em um grafo bipartido balanceado.	8
2.6	O custo de se associar $u_i \in V_1(G_1)$ para $v_j \in V_2(G_2)$ é dado pelo custo c_{ij} associado à aresta $(u_i, v_j) \in E(G_3)$	8
2.7	Comparação entre triangulação e uma coleção qualquer [15].	12
2.8	Triangulação de Delaunay incremental a partir de um conjunto de pontos [30].	13
2.9	Diagrama de Voronoi a partir da Triangulação de Delaunay para um conjunto de pontos [30].	13
2.10	A deformação estrutural aplicada no grafo modelo com a substituição de v_m pelo vértice u_i do grafo de entrada.	15
2.11	Grade log-polar com 5 <i>bins</i> para o logaritmo da distância e 12 <i>bins</i> para o ângulo θ [4].	16
2.12	Diferentes maneiras de se definir um <i>shape context</i> . As numerações indicam as divisões dos <i>bins</i> . A notação P_b^a representa a divisão dos ângulos em a grupos e das distâncias em b grupos [45].	16
3.1	Processo geral para reconhecimento de expressões matemáticas [7].	20
3.2	Hierarquia dos elementos de uma expressão matemática manuscrita <i>online</i>	21
3.3	Similaridade de símbolos matemáticos [51].	22
3.4	Casamento elástico (<i>Elastic Matching</i>) [51].	23
3.5	<i>Dynamic Time Warping</i> entre traços [21].	23
3.6	Árvore estrutural hierárquia da expressão $a^{i^2+2i+1}-b_1^2$ com a distinção entre terminais e sub-estruturas <i>superscript</i> e <i>subscript</i> [50].	24
3.7	Tipos de relações entre símbolos [50].	24
3.8	Uma expressão com diferentes interpretações: $ax + b$, $aX + b$, $a^x + b$, $axtb$, etc. [27].	24
3.9	<i>ExpressMatch: bounding box</i> ao redor dos símbolos durante a escrita [1].	28
3.10	Processo de rotulação de símbolos baseado em agrupamentos de segmentos [26].	30

4.1	Normalização do grafo de entrada com base no grafo modelo. Para efeitos de ilustração, os vértices foram representados pelos respectivos símbolos da expressão associada.	36
4.2	<i>Shape context</i> de um ponto em relação aos outros pontos do símbolo.	37
4.3	Pré-processamento para a expressão $n! - 1$.	37
4.4	<i>Shape context</i> de um símbolo em relação a outros símbolos da expressão.	38
5.1	Exemplos das expressões do conjunto A.	42
5.2	Quantidade de símbolos e instâncias por classe de expressão do conjunto A.	42
5.3	Tela para coleta de expressões no software <i>ExpressMatch</i> [1].	43
5.4	Quantidade de símbolos e instâncias por classe de expressão do conjunto B.	44
5.5	Exemplos de expressões desconsideradas: (a) Incorreta segmentação da expressão $n! - 1$, onde o ponto, “.”, e o sinal de subtração, “-”, foram considerados como sendo traços de um mesmo símbolo (as linhas ao redor dos símbolos apresentam o <i>bounding-box</i> de cada símbolo); (b) O símbolo “+” foi escrito no lugar de “-” (a correta expressão deveria ser $(a\frac{1-t^2}{1+t^2}, \frac{2bt}{1+t^2})$); (c) A posição espacial de x indica a multiplicação $x\sqrt{\frac{a}{b}}$ e a correta expressão seria $\sqrt[3]{\frac{a}{b}}$; (d) Um ponto extra entre os símbolos “a” e “s”.	45
5.6	Associação correta média por casamento obtida com Triangulação de Delaunay e $\beta = \{0.0, 0.25, 0.5, 0.75, 1\}$ para o Conjunto B.	46
5.7	Associação correta média por casamento obtida com grafo completo e $\beta = \{0.0, 0.25, 0.5, 0.75, 1\}$ para o Conjunto B.	47
5.8	Associação correta média global por casamento utilizando as configurações de grafo completo e Triangulação de Delaunay para o grafo modelo.	47
5.9	Associação correta média por casamento obtida com $\gamma = \{0.0, 0.25, 0.5, 0.75, 1\}$ para o Conjunto B.	48
5.10	Associação correta média global por casamento obtida com $\gamma = \{0.0, 0.25, 0.5, 0.75\}$ para o Conjunto B.	49
5.11	Associação correta média global por casamento obtida com $\alpha = [0.0, 1.0]$ para o Conjunto B, com $\beta = 1.0$ e $\gamma = 0.25$.	50
5.12	Associação correta média por casamento obtida com $\alpha = \{0.2, 0.3, 0.4\}$ para o Conjunto B, com $\beta = 1.0$ e $\gamma = 0.25$.	50
5.13	Comparação entre a associação correta média por casamento obtida com custo combinado, vértice e aresta.	51
5.14	Desvio padrão da associação correta média obtido com o custo de aresta ($\beta = 1.0$).	52
5.15	Desvio padrão da associação correta média obtido com o custo de vértice ($\gamma = 0.25$).	53
5.16	Desvio padrão da associação correta média obtido com o custo combinado ($\alpha = 0.3$, $\beta = 1.0$ e $\gamma = 0.25$).	54
5.17	Expressão da classe 27 onde o método de casamento obteve maior desempenho.	59
5.18	Expressões da classe 27 onde o método de casamento obteve menor desempenho como modelo.	59

5.19	Casamento de expressões da classe 27.	60
5.20	Casamento de uma expressão da classe 27 com 13 (65%) símbolos associados corretamente.	61
5.21	Expressões da classe 39.	61
5.22	Expressões da classe 46.	61
5.23	Casamento de expressões da classe 39 com 12 (52,17%) símbolos mapeados corretamente.	62
5.24	Casamento de expressões da classe 46 com 27 (87,09%) símbolos mapeados corretamente.	62
5.25	Expressões da classe 51.	63
5.26	Expressões da classe 52.	63
5.27	Expressões da classe 56.	64
5.28	Associação correta média global por casamento obtida com $\alpha = [0.0, 1.0]$ para o Conjunto A.	64
A.1	Exemplos de expressões do conjunto de dados B (Classes 01 a 13).	67
A.2	Exemplos de expressões do conjunto de dados B (Classes 14 a 25).	68
A.3	Exemplos de expressões do conjunto de dados B (Classes 26 a 37).	69
A.4	Exemplos de expressões do conjunto de dados B (Classes 38 a 48).	70
A.5	Exemplos de expressões do conjunto de dados B (Classes 49 a 56).	71

Lista de Tabelas

5.1	Transcrições desconsideradas do Conjunto B	44
5.2	Resultados dos casamentos para as classes de expressão 1 ~ 28, obtidos com o custo combinado ótimo ($\alpha = 0.3$, $\beta = 1.0$ e $\gamma = 0.25$)	55
5.3	Resultados dos casamentos para as classes de expressão 29 ~ 56, obtidos com o custo combinado ótimo ($\alpha = 0.3$, $\beta = 1.0$ e $\gamma = 0.25$)	56
5.4	Resultados do casamento categorizados por intervalos de desempenho obtido com o custo combinado ótimo ($\alpha = 0.3$)	58

Capítulo 1

Introdução

Com o advento das tecnologias com interfaces *pen-based* e *touch-screen*, houve um aumento do interesse no campo de reconhecimento da escrita nas últimas décadas [2, 17, 42, 51]. Além do teclado virtual e do teclado físico presente em alguns modelos, a maioria dos atuais *tablets*, *Personal Digital Assistants* (PDAs) e *SmartPhones* possibilitam diferentes formas de entrada de dados. Uma dessas formas é a escrita.

A entrada de elementos diferentes de texto tais como expressões matemáticas, fórmulas, diagramas, gráficos, etc. pode ser mais facilmente realizada por meio da escrita do que pelos métodos convencionais como teclado e mouse. Nestes últimos, exige-se que o usuário domine alguma linguagem específica própria para expressar esses objetos, como o $\text{\LaTeX}$, ou então selecione símbolos ou primitivas de, por exemplo, um menu e os componha manualmente.

Nos últimos anos, vários trabalhos vêm se dedicando ao estudo de soluções para o reconhecimento da escrita. Entre os tipos de escrita em estudo, um que se destaca são as de expressões matemáticas [3, 51]. O reconhecimento de expressões matemáticas tem uma grande importância prática e possui diversas utilidades, tais como facilitar a entrada de expressões em sistemas que realizam processamento algébrico; facilitar a produção (tipografiação) de expressões matemáticas em documentos gerados por computador (como $\text{\LaTeX}$); fazer leitura de documentos para pessoas portadoras de deficiência visual [47]; e fazer o reconhecimento de textos técnicos e científicos (que geralmente incluem fórmulas matemáticas). Em geral, as pesquisas na área combinam as vantagens naturais da escrita manuscrita com a capacidade de processamento dos computadores [29].

O problema de reconhecimento de expressões matemáticas inclui geralmente três subproblemas: segmentação dos símbolos, reconhecimento dos símbolos e análise estrutural da expressão [20]. Dado um conjunto de traços resultante da escrita, a segmentação consiste em agrupá-los em símbolos; o reconhecimento de símbolos identifica o símbolo representado por cada grupo a partir de características extraídas do mesmo; e a análise estrutural determina a relação espacial entre os símbolos, de forma a reconhecer a semântica associada aos mesmos e a expressão em si.

O problema pode ainda ser classificado em *offline* ou *online*. No primeiro, apenas uma imagem da expressão é disponível, enquanto no segundo informações temporais associadas aos traços da escrita são disponíveis. No caso de reconhecimento *online*, que é o contexto considerado neste trabalho,

as técnicas podem explorar as informações temporais e, portanto, tendem a apresentar melhor desempenho do que o reconhecimento *offline*.

Mesmo no caso *online* o reconhecimento não é fácil. Mouchere et. al. [31] descreve o desempenho de cinco sistemas de reconhecimento que participaram da competição CROHME2011 considerando os subproblemas acima apresentados e um dado conjunto de dados de expressões matemáticas *online*. Os dois melhores sistemas apresentaram, respectivamente, taxas de acerto de 88,07% e 87,56% para a segmentação; 92,22% e 91,67% para o reconhecimento de símbolos; e 29,28% e 40,88% para o reconhecimento de expressões. Neste último, a maior dificuldade encontrada é a presença de ambiguidades no entendimento da estrutura bidimensional da expressão.

1.1 Motivação

Uma das dificuldades que tem sido recentemente apontada na área de reconhecimento de expressões matemáticas é a quase inexistência de base de dados pública [3,23]. As bases existentes são, em geral, limitadas quanto ao tipo de símbolos, expressões, quantidade de pessoas e de amostras. Muitas vezes, são disponibilizadas apenas mediante registro.

Verifica-se na literatura que, em geral, os autores geram seus próprios conjuntos de dados para desenvolver e avaliar as técnicas de reconhecimento. Isto tende a gerar técnicas restritas a um subdomínio (nem sempre muito claro), o que representa um viés na avaliação e dificulta a comparação entre diferentes técnicas. A existência de bases de dados expressivas, estatisticamente representativas e publicamente disponíveis facilitaria os trabalhos de desenvolvimento, avaliação e comparação das técnicas [26]. Isso certamente ajudaria a agilizar os avanços nessa área.

Esses fatos levaram-nos à conclusão de que para proceder com qualquer investigação nessa área, antes de mais nada, seria necessário gerar um conjunto de dados que pudesse ser usado para o desenvolvimento e avaliação das técnicas de reconhecimento. Uma vez que havia um interesse inicial em desenvolver classificadores para o reconhecimento de símbolos matemáticos manuscritos, tornou-se necessária a criação de um conjunto com amostras de tais símbolos, estes escritos por diferentes pessoas.

Geralmente, tais amostras são obtidas solicitando-se às pessoas que escrevam diferentes símbolos individualmente e, muito frequentemente, repetidas vezes. Porém essa tarefa, do ponto de vista de quem escreve os símbolos, é monótona, demorada e cansativa. Por outro lado, escrever algumas expressões em vez de símbolos isolados certamente é uma atividade menos monótona.

Motivado por essa constatação, decidiu-se que uma abordagem interessante para realizar a coleta de amostras de símbolos seria por meio da coleta de amostras de expressões matemáticas completas. Os símbolos seriam, então, extraídos das expressões e anotados individualmente com seus respectivos rótulos (classes), processo também conhecido como rotulação. Devido a grande quantidade de símbolos e expressões, realizar tal anotação manualmente é, também, uma tarefa monótona, demorada, cansativa e sujeita a erros. A existência de sistemas para facilitar a anotação eliminaria ou,

pelo menos, minimizaria a atuação manual no processo.

1.2 Objetivos

O objetivo deste trabalho é tratar o problema de rotular os símbolos de uma expressão manuscrita. Mais especificamente, os objetivos são:

1. desenvolver um método para casamento de expressões matemáticas manuscritas no qual símbolos de uma expressão transcrita por um usuário são associados aos correspondentes símbolos (previamente identificados) da expressão modelo;
2. implementar e validar o método;
3. criar um conjunto de símbolos manuscritos anotados com *ground-truth* usando o método proposto.

1.3 Principais contribuições e resultados

Dentre as principais contribuições, podemos citar as seguintes:

- foi desenvolvido um método de casamento de expressões, formulado como um problema de associação simples. As expressões matemáticas são representadas por grafos e para os custos de associação é proposta a combinação de características locais (características dos símbolos) e características estruturais globais (relação espacial entre os símbolos);
- foi desenvolvido um software juntamente com o aluno Frank D. J. Aguilar [1], denominado *ExpressMatch*, que possibilita a definição de expressões modelo, coleta de amostras por meio de transcrições das expressões modelo, a anotação dos símbolos usando o método de casamento proposto, correção da anotação e a exportação do conjunto de símbolos para um arquivo. A coleta das expressões de um dos conjuntos de dados utilizados neste trabalho foi realizada com esse software;
- resultados preliminares foram publicados no artigo *Automatic Labeling of Handwritten Mathematical Symbols via Expression Matching* [18], apresentado no *8th IAPR - TC-15 Workshop on Graph-based Representations in Pattern Recognition*, em maio de 2011;

1.4 Organização do texto

Este texto está organizado da seguinte forma. No capítulo 2, apresenta-se os conceitos e definições básicas utilizados neste trabalho. No capítulo 3, apresenta-se uma visão panorâmica sobre o problema de reconhecimento de expressões matemáticas, assim como a descrição da abordagem proposta e

trabalhos relacionados encontrados na literatura. No capítulo 4, descreve-se a técnica para casamento de expressões matemáticas em termos de emparelhamento de grafos bipartidos. No capítulo 5, apresenta-se a metodologia e os resultados experimentais obtidos com o método. Finalmente, no capítulo 6, apresenta-se as conclusões obtidas neste trabalho.

Capítulo 2

Preliminares

Neste capítulo, são apresentados alguns dos conceitos básicos que serão utilizados neste trabalho. Os tópicos abordados neste capítulo não necessariamente estão relacionados diretamente uns aos outros.

2.1 Grafos

Definição: um grafo G pode ser definido como um par ordenado (V, E) , constituído de um conjunto V de vértices (também chamados de nodos ou pontos), um conjunto E de arestas, disjunto de V , e uma função de incidência ψ_G que associa cada aresta de G a um par não ordenado (não necessariamente distintos) de vértices de G [6].

Vértices adjacentes: se dois vértices $u, v \in V$ são conectados por uma aresta $e \in E$, então denota-se $e = (u, v)$ e dizemos que ambos os vértices são adjacentes ou vizinhos entre si. Neste caso, u e v são denominados extremos da aresta e .

Grafo finito: quando V e E são conjuntos finitos, G constitui um grafo finito.

Grafo direcionado e não-direcionado: caso todas as arestas do grafo possuam direção, o grafo é dito direcionado. Nos grafos direcionados, a relação entre seus vértices não é simétrica, isto é, a aresta (u, v) é diferente da aresta (v, u) . Por outro lado, se a aresta não possui direção, então a mesma é denominada não-direcionada, indicando, simplesmente, a existência de uma relação entre dois vértices [5]. Um grafo G contendo apenas arestas não-direcionadas é dito não-direcionado.

Grafo relacional com atributo: um *Attributed Relational Graph* (ARG) pode ser considerado uma extensão da definição de um grafo comum. É definido como $G = (V, E, A)$, onde V é o conjunto de vértices, E é o conjunto de arestas, e A é o conjunto de atributos, i.e, atributos unários a_i associados a vértices $v_i \in V$ e atributos binários a_{ij} associados a arestas $e = (v_i, v_j) \in E$ [56].

Grafo completo: um grafo G é denominado completo quando há uma aresta $(u, v) \in E$ entre quaisquer dois vértices u e v em V . Alguns exemplos de grafos completos são apresentados na Figura 2.1.

Figura 2.1: Exemplos de grafos completos.

Grafo planar: um grafo é denominado planar se ele pode ser representado (desenhado) no plano sem cruzamento de arestas. Um grafo planar divide o plano em regiões por meio de suas arestas. Cada uma dessas regiões é denominada de face do grafo.

Grafo bipartido: um grafo $G = (V, E)$ é bipartido se existe uma partição do conjunto de todos os vértices $V(G)$ em dois subconjuntos X e Y , tal que $V(G) = X \cup Y$, $X \cap Y = \emptyset$ e nenhuma aresta do grafo tenha ambas as extremidades em X ou ambas as extremidades em Y . Denotamos, então, G como $G[X, Y]$, sendo X e Y suas partes. Um exemplo de grafo bipartido pode ser visto na Figura 2.2.

Figura 2.2: Exemplo de um grafo bipartido.

Grafo bipartido balanceado: um grafo $G[X, Y]$ é um grafo bipartido balanceado se $|X| = |Y|$, isto é, os dois subconjuntos de vértices tem igual cardinalidade.

Grafo bipartido completo: um grafo bipartido $G[X, Y]$ é chamado de completo quando todos

os vértices de X estão ligados a todos os vértices de Y . A Figura 2.3 apresenta um exemplo de grafo bipartido completo.

Figura 2.3: Um grafo bipartido completo com três vértices em cada parte.

Emparelhamento: um emparelhamento M no grafo $G = (V, E)$ é um conjunto de arestas não adjacentes entre si, isto é, quaisquer duas arestas nesse conjunto não possuem vértices em comum. Logo, M é dotado da seguinte propriedade: todo vértice de G incide em no máximo um elemento de M . Dizemos que $v \in V(G)$ é coberto por M se v for extremo de alguma aresta de M .

Emparelhamento maximal: um emparelhamento M é maximal se não existe um emparelhamento M' que contenha o próprio M , isto é, M é maximal se não existe uma aresta e que não pertence a M tal que $M + e$ também é um emparelhamento.

Emparelhamento máximo: um emparelhamento M é máximo se não existe um emparelhamento M' tal que $|M'| > |M|$. A Figura 2.4 apresenta as diferenças entre um emparelhamento maximal e emparelhamento máximo em um mesmo grafo.

Figura 2.4: Diferenças entre um emparelhamento maximal e emparelhamento máximo em um mesmo grafo [9].

Emparelhamento perfeito: Um emparelhamento M no grafo G é perfeito se cobre todo vértice $v \in V(G)$.

2.2 Emparelhamento de grafos bipartidos balanceados

O problema de emparelhamento de grafos bipartidos balanceados consiste em, dado um grafo bipartido balanceado $G[X, Y] = (V, E)$ não-nulo, encontrar um emparelhamento máximo em G . Esse problema também pode ser formulado com custos nas arestas. Neste caso, o emparelhamento ótimo é o emparelhamento máximo M de menor custo, isto é, aquele cuja soma dos custos das arestas de M é mínima em E .

Figura 2.5: Emparelhamento em um grafo bipartido balanceado.

2.2.1 Associação de dois grafos relacionais com atributos como emparelhamento de grafo bipartido

O problema de associação de dois grafos relacionais com atributos (ARGs), $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ tal que $|V_1| = |V_2|$, quando existe interesse em se estabelecer associação um-para-um entre vértices de um deles com vértices do outro, pode ser visto como um problema de emparelhamento de grafo bipartido balanceado. Nessa formulação, considera-se um terceiro grafo $G_3 = G[V_1, V_2] = (V, E)$, com $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ e $E = \{(u, v) : u \in V_1 \text{ e } v \in V_2\}$. Então, o custo c_{ij} de se associar $u_i \in V_1(G_1)$ para $v_j \in V_2(G_2)$ pode ser associado à aresta $(u_i, v_j) \in E(G_3)$. A Figura 2.6 ilustra essa representação.

Figura 2.6: O custo de se associar $u_i \in V_1(G_1)$ para $v_j \in V_2(G_2)$ é dado pelo custo c_{ij} associado à aresta $(u_i, v_j) \in E(G_3)$.

O problema de emparelhamento de grafos bipartidos balanceados, utilizando ou não atributos, é conhecido como Problema de Associação (também conhecido como Problema de Designação). Um algoritmo conhecido para a solução deste problema é o Algoritmo Húngaro.

2.2.2 Algoritmo Húngaro

O Algoritmo Húngaro é um método de otimização combinatória que soluciona o Problema de Associação em um tempo polinomial $O(n^3)$ [4], no qual n é a quantidade total de elementos. Originalmente escrito por Kuhn [24] e revisado por Munkres [32], este algoritmo é também conhecido como Algoritmo de Kuhn-Munkres ou Casamento de Munkres.

Dado um grafo bipartido completo balanceado $G[X, Y]$, no qual $n = |X| + |Y|$, $X \cap Y = \emptyset$ e que cada aresta (x_i, y_j) , tal que $x_i \in X$ e $y_j \in Y$, tem um custo associado $c_{ij} = c(x_i, y_j)$, o objetivo do algoritmo é encontrar um emparelhamento ótimo, isto é, um emparelhamento perfeito com custo mínimo. Seja a variável de decisão

$$\chi_{ij} = \begin{cases} 1, & \text{se a aresta } (x_i, y_j) \text{ pertencer ao emparelhamento;} \\ 0, & \text{caso contrário.} \end{cases}$$

Então, matematicamente o problema pode ser formulado da seguinte maneira [46]:

$$\begin{cases} \text{minimizar } \sum_i \sum_j c_{ij} \chi_{ij} \\ \text{sujeito a:} \\ \sum_i \chi_{ij} = 1, \forall j \in \{1, 2, \dots, n\}; \\ \sum_j \chi_{ij} = 1, \forall i \in \{1, 2, \dots, n\}; \\ \chi_{ij} \in \{0, 1\}, \forall i \in \{1, 2, \dots, n\} \text{ e } \forall j \in \{1, 2, \dots, n\}. \end{cases} \quad (2.1)$$

O Algoritmo Húngaro recebe como entrada a matriz C de custos $c(x_i, y_j)$ de dimensão $n \times n$, que é simétrica, e realiza sucessivas transformações em C para encontrar a associação ótima. Basicamente, consiste nas quatro fases a seguir:

1. Fase 1: Redução de linhas e colunas da matriz C . O objetivo é facilitar a identificação das melhores opções de associação.
 - 1.1 Para cada linha da matriz, subtraia o menor elemento da mesma.
 - 1.2 Para cada coluna da matriz, subtraia o menor elemento da mesma.
2. Fase 2: Identificação de uma solução ótima. Nessa fase, tenta-se encontrar uma associação que utilize as associações de cada elemento da linha de C com cada elemento da coluna de C .
 - 2.1 Encontre a quantidade mínima de linhas que cubram todos os zeros da matriz;
 - 2.2 Se a quantidade de linhas no conjunto for menor do que n , vá para a Fase 3, senão vá para a Fase 4.

3. Fase 3: Alterações adicionais na matriz C .

- 3.1 Seja m o menor elemento da matriz não coberto pelas linhas;
- 3.2 Para cada elemento não coberto pelas linhas, subtraia m ;
- 3.3 Para cada elemento coberto por duas linhas simultaneamente, adicione m ;
- 3.4 Manter os demais elementos como estão;
- 3.5 Volte para a Fase 2.

4. Fase 4: Solução ótima. Tem-se n zeros independentes na matriz.

- 4.1 Encontre a associação com n zeros na matriz.

Para exemplificar o funcionamento do algoritmo, suponha que a matriz 4×4 abaixo representa uma matriz de custos C .

$$\begin{bmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \\ 125 & 95 & 90 & 105 \\ 45 & 110 & 95 & 115 \end{bmatrix}$$

Na Fase 1, para cada linha da matriz, o menor elemento é subtraído de todos os elementos da mesma.

$$\begin{bmatrix} 90 & \boxed{75} & 75 & 80 \\ \boxed{35} & 85 & 55 & 65 \\ 125 & 95 & \boxed{90} & 105 \\ \boxed{45} & 110 & 95 & 115 \end{bmatrix} \approx \begin{bmatrix} 15 & 0 & 0 & 5 \\ 0 & 50 & 20 & 30 \\ 35 & 5 & 0 & 15 \\ 0 & 65 & 50 & 70 \end{bmatrix}$$

Ainda na Fase 1, para cada coluna da matriz, o menor elemento é subtraído de todos os elementos da mesma.

$$\begin{bmatrix} 15 & \boxed{0} & \boxed{0} & \boxed{5} \\ \boxed{0} & 50 & 20 & 30 \\ 35 & 5 & 0 & 15 \\ 0 & 65 & 50 & 70 \end{bmatrix} \approx \begin{bmatrix} 15 & 0 & 0 & 0 \\ 0 & 50 & 20 & 25 \\ 35 & 5 & 0 & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix}$$

Na Fase 2, a quantidade mínima de linhas que cobrem todos os zeros da matriz é obtida. Verifica-se nesse exemplo que a quantidade de linhas é igual a 3, menor do que $n = 4$.

$$\begin{bmatrix} \cancel{15} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 50 & 20 & 25 \\ 35 & 5 & 0 & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix}$$

Na Fase 3, sendo 5 o menor elemento não coberto pelas linhas, tem-se a seguinte matriz após as etapas:

$$\begin{bmatrix} 20 & 0 & 5 & 0 \\ 0 & 45 & 20 & 20 \\ 35 & 0 & 0 & 5 \\ 0 & 60 & 50 & 60 \end{bmatrix}$$

Novamente à Fase 2, verifica-se que a quantidade mínima de linhas em C é igual a 3, menor do que $n = 4$.

$$\begin{bmatrix} \cancel{20} & \cancel{0} & \cancel{5} & \cancel{0} \\ 0 & 45 & 20 & 20 \\ \cancel{35} & 0 & 0 & 5 \\ 0 & 60 & 50 & 60 \end{bmatrix}$$

Na Fase 3, sendo 20 o menor elemento não coberto pelas linhas, tem-se a seguinte matriz após as etapas:

$$\begin{bmatrix} 40 & 0 & 5 & 0 \\ 0 & 25 & 0 & 0 \\ 55 & 0 & 0 & 5 \\ 0 & 40 & 30 & 40 \end{bmatrix}$$

Na Fase 2, desta vez verifica-se que a quantidade mínima de linhas em C é igual a $n = 4$.

$$\begin{bmatrix} \cancel{40} & \cancel{0} & \cancel{5} & \cancel{0} \\ \cancel{0} & \cancel{25} & \cancel{0} & \cancel{0} \\ \cancel{55} & \cancel{0} & \cancel{0} & \cancel{5} \\ \cancel{0} & \cancel{40} & \cancel{30} & \cancel{40} \end{bmatrix}$$

Finalmente, observa-se que há 4 zeros independentes na matriz, o que representa a solução ótima. No entanto, a solução ótima não é única nesse caso, como pode-se observar a seguir.

$$\begin{bmatrix} 40 & 0 & 5 & \boxed{0} \\ 0 & 25 & \boxed{0} & 0 \\ 55 & \boxed{0} & 0 & 5 \\ \boxed{0} & 40 & 30 & 40 \end{bmatrix} \quad \begin{bmatrix} 40 & \boxed{0} & 5 & 0 \\ 0 & 25 & 0 & \boxed{0} \\ 55 & 0 & \boxed{0} & 5 \\ \boxed{0} & 40 & 30 & 40 \end{bmatrix}$$

2.3 Triangulação de Delaunay

Um complexo simplicial é uma coleção de triângulos que só se interceptam em vértices e arestas. Logo, dados quaisquer dois triângulos do complexo simplicial, uma das quatro situações ocorre: eles são disjuntos, eles têm exatamente um vértice em comum, eles têm exatamente uma aresta em comum ou eles são iguais [10].

Seja P um conjunto de pontos desconectados no plano. Uma maneira de organizar P é definir sobre ele uma relação de vizinhança, isto é, conectar os pontos segundo algum critério de forma a estabelecer uma subdivisão do espaço que ocupam.

O fecho convexo de P é definido como o menor polígono convexo que contém todos os pontos. Neste caso, cada ponto do conjunto ou é um vértice desse polígono (o fecho) ou está no seu interior [10].

A triangulação pode ser estabelecida como a decomposição do fecho convexo de P em um complexo simplicial. A Figura 2.7 apresenta uma comparação entre uma triangulação e uma coleção que não satisfaz as situações de um complexo simplicial.

Figura 2.7: Comparação entre triangulação e uma coleção qualquer [15].

Um processo de triangulação largamente conhecido é a Triangulação de Delaunay. Proposta por Boris Delaunay, a Triangulação de Delaunay é um tipo de triangulação no qual os triângulos são formados progressivamente a partir do seguinte critério:

Definição 2.1 (Critério de Delaunay) *Um determinado triângulo t faz parte da Triangulação de Delaunay T se, e somente se, o círculo formado pelos vértices do triângulo não contém nenhum outro vértice de T em seu interior.*

No entanto, essa regra é fraca, pois podem existir 4 vértices co-circulares, a , b , c e d , sem existir outros vértices no interior do círculo, o que faria com que os 4 triângulos, abc , abd , acd e bcd , satisfizessem o critério, mas somente 2 deles poderiam fazer parte da triangulação. Isso faz com que a Triangulação de Delaunay não seja única [15].

Se assumirmos que o conjunto de pontos não contém 4 pontos na situação acima, podemos garantir que a triangulação é única, e o critério, além de condição necessária, passa a ser também

uma condição suficiente [15].

A partir de um conjunto P de pontos, a Triangulação de Delaunay pode ser feita incrementalmente, com a formação de círculos e a subsequente adição de arestas. Esse processo pode ser visto na Figura 2.8.

Figura 2.8: Triangulação de Delaunay incremental a partir de um conjunto de pontos [30].

Uma característica peculiar da Triangulação de Delaunay é que, em um espaço bidimensional, sua estrutura dual corresponde ao Diagrama de Voronoi. Neste caso, os centróides das regiões de Voronoi correspondem aos vértices da Triangulação de Delaunay e as fronteiras de cada região são linhas perpendiculares ao ponto médio de cada aresta da triangulação. A Figura 2.9 apresenta esse relacionamento.

Figura 2.9: Diagrama de Voronoi a partir da Triangulação de Delaunay para um conjunto de pontos [30].

O Diagrama de Voronoi é um método que tem extrema utilidade em reconhecimento de pa-

drões para delimitar as áreas de categorias ou classes em uma representação geométrica de suas distribuições.

2.4 Custo de deformação de aparência e estrutural

Objetos presentes em imagens podem ser representados por ARGs, nos quais os vértices representam partes dos objetos e as arestas expressam as relações entre as partes. A associação entre o ARG de um objeto modelo e outro ARG representando outro objeto similar é uma abordagem de reconhecimento de padrões para identificar as partes do objeto similar, associando-as às correspondentes partes do objeto modelo [33].

Os custos de associação devem, de alguma forma, capturar aspectos de aparência de cada parte do objeto (por exemplo, forma, tamanho, e cor) e aspectos estruturais do objeto. Noma et. al. [33] propuseram um custo, aqui denominado de custo de deformação, que contempla esses aspectos.

Seja G_M o ARG de um objeto modelo e G_I o ARG representando outro objeto similar ao modelo. A ideia básica do custo consiste em expressar a deformação sofrida por G_M se o vértice $v_m \in V(G_M)$ for trocado por um vértice $u_i \in V(G_I)$. Em sua forma geral, ele é dado pela seguinte expressão:

$$c(u_i, v_m) = \alpha c_v(u_i, v_m) + (1 - \alpha) c_e(u_i, v_m) \quad (2.2)$$

onde $v_m \in V(G_M)$, $u_i \in V(G_I)$, c_v representa o custo associado ao vértice (aspectos de aparência), c_e representa o custo associado à aresta (aspectos estruturais) e α é um número real entre 0 e 1, usado para ponderar a importância entre os dois termos. Sendo $0 \leq \alpha \leq 1$, é importante que c_v e c_e estejam normalizados, isto é, $0 \leq c_v, c_e \leq 1$.

As deformações na aparência são capturadas por c_v e as deformações estruturais por c_e . Por exemplo, enquanto c_v expressa particularidades quanto à forma e formato do objeto representado, c_e pode expressar valores que exprimem informações como comprimento e orientação das arestas. A ideia da deformação estrutural é ilustrada na Figura 2.10.

Para cada problema de associação entre ARGs, os custos c_v e c_e apropriados devem ser utilizados de acordo com o seu contexto.

Figura 2.10: A deformação estrutural aplicada no grafo modelo com a substituição de v_m pelo vértice u_i do grafo de entrada.

2.5 Shape context

Proposto por Belongie et. al. [4], o *shape context* é um descritor que serve para mensurar a similaridade entre duas formas. Dado um conjunto de pontos do contorno do objeto, o objetivo é, para cada ponto desse conjunto, associar informações sobre as posições relativas dos demais pontos. É natural esperar que dois objetos com formas similares sejam também similares com respeito a esse tipo de informação. Como, em geral, a quantidade de pontos considerados pode ser grande, essas informações são armazenadas em um histograma, que define o chamado *shape context*. Tal histograma é composto por regiões denominadas *bins*. Para a criação dos *bins*, considera-se o espaço log-polar relativo ao ponto em questão, descrito a seguir.

Basicamente, o espaço log-polar é um sistema de coordenadas em duas dimensões, no qual um ponto p é identificado por dois números: um referente ao logaritmo da distância até o ponto de referência e outro número referente ao ângulo formado entre p e esse ponto de referência. Um diagrama de *bins* no espaço log-polar é mostrado na Figura 2.11.

Os *bins* do *shape context* podem ser divididos em grupos, desde que sejam uniformemente distribuídos e sem sobreposição. Na Figura 2.12(a), as distâncias e ângulos são divididos pela metade e na Figura 2.12(b) apenas os ângulos foram divididos. A Figura 2.12(c) apresenta dois intervalos de distância e quatro quadrantes, totalizando 8 *bins*. Essa divisão em *bins* torna o descritor mais sensível para as coordenadas de pontos mais próximos do que os pontos mais afastados [4].

Seja p_i um ponto do contorno de uma forma. O *shape context* (histograma) relacionado a p_i é

Figura 2.11: Grade log-polar com 5 *bins* para o logaritmo da distância e 12 *bins* para o ângulo θ [4].

Figura 2.12: Diferentes maneiras de se definir um *shape context*. As numerações indicam as divisões dos *bins*. A notação P_b^a representa a divisão dos ângulos em a grupos e das distâncias em b grupos [45].

definido da seguinte maneira:

$$hp_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\} \quad (2.3)$$

no qual q são os outros pontos do contorno diferente de p_i e $bin(k)$ é o k -ésimo *bin* do histograma.

Se duas formas forem bastante similares, é de se esperar que para cada ponto de uma das formas exista um correspondente ponto na outra forma com *shape context* muito similar. Belongie et. al. [4] definem também uma medida para comparar o *shape context* de dois pontos p_i e q_j :

$$C_{ij} \equiv C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{[hp_i(k) - hq_j(k)]^2}{hp_i(k) + hq_j(k)} \quad (2.4)$$

na qual K é a quantidade total de *bins* do *shape context* e hp_i e hq_j são, respectivamente, os histogramas associados a p_i e q_j . Caso $hp_i(k) = hq_j(k) = 0$, o correspondente termo não é considerado no somatório.

Para mensurar a similaridade entre duas formas, não é suficiente apenas associar cada ponto da primeira forma ao ponto mais similar da segunda forma. Deve-se encontrar a melhor associação um-para-um entre os pontos das duas formas, que pode ser caracterizado como a associação que

minimiza o custo total dado por:

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}) \quad (2.5)$$

na qual π indica uma permutação [4]. A associação de custo mínimo pode ser obtido a partir do Algoritmo Húngaro, apresentado na seção 2.2.2.

Capítulo 3

Reconhecimento de expressões matemáticas manuscritas

O reconhecimento de expressões matemáticas manuscritas difere significativamente de outras formas de escrita e tem sido estudado a mais de 30 anos [51]. Durante esse tempo, diversas técnicas foram desenvolvidas para o reconhecimento de expressões matemáticas *online*.

Esse capítulo apresenta uma introdução sobre escrita *online*, um panorama sobre o reconhecimento de expressões matemáticas, descrição do problema a ser tratado neste trabalho juntamente com uma visão geral da abordagem proposta e trabalhos relacionados.

3.1 Escrita online

No contexto de reconhecimento de escrita, o termo escrita *online* é utilizado em contraposição à escrita *offline*. No caso da escrita *offline*, o reconhecimento é realizado a partir de uma imagem da escrita, obtida, por exemplo, por meio de um scanner de mesa ou mesmo por ilustrações impressas. No caso da escrita *online*, os dados da escrita são obtidos por meio de dispositivos tais como os *tablets*, que capturam não apenas a distribuição espacial dos traços da escrita, como também a informação temporal associada a eles [34].

Alguns tipos de escrita que vêm recebendo atenção em termos de reconhecimento são a escrita matemática manuscrita, incluindo números, expressões matemáticas e diagramas em geral. Em todos esses casos, os elementos atômicos da captura da escrita *online* são pontos (coordenadas no espaço 2D) juntamente com a informação do instante da captura. Em geral, as primitivas da captura são os traços, sequência de pontos gerados entre um *pen-down* (ato da caneta tocar a superfície do dispositivo de captura) e um *pen-up* (subsequente ato no qual a caneta se afasta da superfície do dispositivo de captura).

Para propósitos de análise, os pontos ou traços capturados precisam ser agrupados de forma que cada grupo represente uma unidade semântica. Os agrupamentos podem ser circunscritos por um retângulo, denominado *bounding-box*, que corresponde ao menor retângulo que contém todos os

pontos do agrupamento. O ponto de coordenadas (x, y) centrais desse retângulo é denominado de centróide.

3.2 Panorama sobre o reconhecimento de expressões matemáticas manuscritas

Geralmente, os trabalhos da área de reconhecimento de expressões matemáticas dividem o problema em três etapas [7]: segmentação de caracteres, reconhecimento de caracteres, e análise estrutural, conforme ilustrado na Figura 3.1. Essas três etapas podem ser tratadas de forma integrada, mas muitos trabalhos na literatura as tratam de forma independente [2,20] ou então concentram-se na investigação de apenas uma dessas etapas. Outros trabalhos utilizam abordagens nas quais essas etapas não são evidentes, ao realizar, por exemplo, a análise estrutural antes do reconhecimento de símbolos [44], ou realizar a segmentação e o reconhecimento simultaneamente [2,14].

Figura 3.1: Processo geral para reconhecimento de expressões matemáticas [7].

3.2.1 Segmentação

Quando uma expressão matemática é escrita em um dispositivo como o *tablet*, uma sequência de pontos é obtida de maneira *online* (ver seção 3.1), formando traços. A etapa de segmentação de símbolos é responsável por agrupar os traços de forma que cada grupo corresponda a um símbolo na expressão [2, 40]. Em geral, esses agrupamentos resultam em uma representação hierárquica, no qual grupos de pontos constituem traços, grupos de traços constituem símbolos e, finalmente, grupos de símbolos constituem expressões matemáticas. O reconhecimento de expressões matemáticas *online* pode, portanto, explorar essa organização hierárquica dos dados de diferentes formas. A Figura 3.2 apresenta essa organização.

Figura 3.2: Hierarquia dos elementos de uma expressão matemática manuscrita *online*.

Furukawa et. al. [12] listam os três tipos principais de informação em uma segmentação de símbolos:

Física: características geométricas e temporais, como largura e altura dos segmentos, assim como o tempo de escrita de cada um;

Estatística: frequência de coocorrência dos caracteres concatenados, probabilidades de transição, etc.

Linguística: uso de modelos de linguagens ou análises morfológicas.

Em reconhecimento de expressões matemáticas *online*, a etapa de segmentação é relativamente simples comparada ao caso *offline*, uma vez que é possível utilizar a informação temporal dos traços. No momento da coleta da expressão matemática, os usuários podem, por exemplo, serem orientados a somente iniciar a escrita do próximo símbolo depois de um certo intervalo de tempo após o término do último traço do símbolo anterior [18].

A segmentação pode se tornar mais complexa quando os traços são escritos sem uma ordem específica ou diversos símbolos são representados por um único traço contínuo (por exemplo, a escrita manuscrita contínua) [12].

3.2.2 Reconhecimento de símbolos

Após agrupar os traços de tal modo que cada agrupamento represente um único símbolo, pode-se aplicar uma técnica para reconhecimento de símbolos a fim de identificá-lo e rotulá-lo.

O reconhecimento de símbolos é um problema complexo pois, primeiramente, há uma variedade de símbolos que podem ser utilizados nas expressões matemáticas, incluindo caracteres, numerais, símbolos gregos, parênteses, operadores, etc [7]. Mais de 220 símbolos são necessários para abranger grande parte das aplicações científicas [12] e um classificador completo deve reconhecer aproximadamente 2000 símbolos [51]. Segundo, existem tipos de símbolos muito similares e geralmente compostos de poucos traços, escritos sem uma ordem específica. Por exemplo, um segmento horizontal pode ser considerado como um sinal de subtração ou um traço de fração. Outros exemplos são apresentados na Figura 3.3 [51].

Figura 3.3: Similaridade de símbolos matemáticos [51].

Há diversas técnicas para o reconhecimento de símbolos matemáticos. Tapia [43] lista as três principais categorias de técnicas para classificação:

Medidas de similaridade: são técnicas baseadas em distância. Geralmente, os símbolos desconhecidos são comparados com modelos de símbolos armazenados e classificados como sendo da mesma classe do modelo mais similar. O casamento elástico ou *Elastic Matching* (Figura 3.4) e *Dynamic Time Warping* (Figura 3.5) são técnicas dessa categoria.

Modelos estatísticos: são técnicas que utilizam modelos estatísticos para a classificação de símbolos, tais como Classificadores Bayesianos, Redes Neurais Artificiais, Máquinas de Vetores-Suporte (*Support Vector Machines*) e Modelos Ocultos de Markov.

Métodos de agrupamento: tentam agrupar variações na escrita de símbolos causadas por diferentes estilos de escrita. Exemplos de técnicas nessa categoria: Mapas Auto-Organizáveis, *k*-Vizinhos Mais Próximos e Agrupamento Hierárquico.

Em geral, os trabalhos voltados ao reconhecimento de símbolos em expressões matemáticas supõem que os símbolos já se encontram segmentados [13, 44, 54].

3.2.3 Análise estrutural

A análise estrutural investiga a posição relativa entre os símbolos na expressão para determinar o papel semântico dos mesmos. Os símbolos são conectados por relações espaciais que consideram suas posições espaciais e tamanhos, e podem ser agrupados para representar um significado específico.

Figura 3.4: Casamento elástico (*Elastic Matching*) [51].

Figura 3.5: *Dynamic Time Warping* entre traços [21].

Por exemplo, letras e números podem ser agrupados para representar um significado próprio (por exemplo, 565.15 e cos), enquanto outros operadores que dominam sub-expressões, tais como $\sum$ e $\int$, podem ser agrupados como uma única unidade (por exemplo, $\sum_{x=0}^{100} x^2$) [20].

A informação estrutural é geralmente organizada de forma hierárquica [51] e pode ser representada por uma árvore estrutural hierárquica [7] como apresentado na Figura 3.6. Nessa árvore, a raiz representa a expressão em si e os nodos representam seus componentes. Esses componentes podem ser simplesmente símbolos ou sub-estruturas intermediárias, tais como sub-expressões que aparecem como *superscript* e *subscript* [20]. As arestas da árvore representam as relações espaciais entre esses componentes [52]. A Figura 3.7 apresenta alguns tipos de relações entre os símbolos de uma expressão.

A análise estrutural é um problema difícil pois envolve fatores que dificultam encontrar com exatidão as relações espaciais entre os símbolos da expressão. Dentre esses fatores, pode-se citar: a ambiguidade do conteúdo manuscrito da expressão [11] (um exemplo de ambiguidade de interpretação é ilustrada na Figura 3.8); formas diferentes de interpretar uma mesma expressão (por exemplo, $a/2b$ pode ser interpretada como $\frac{a}{2}b$ ou $\frac{a}{2b}$ [3]); e podem existir várias formas diferentes que representam expressões equivalentes, como por exemplo em $-(a/b)$, $-\frac{a}{b}$ e $-(\frac{a}{b})$ [3]. Além disso, a notação matemática não é formalmente definida. Existem diferentes comunidades matemáticas e cada uma com sua própria convenção, o que dificulta a elaboração de um método que atenda todas as convenções possíveis [51].

Figura 3.6: Árvore estrutural hierárquica da expressão $a^{i^2+2i+1} - b_1^2$ com a distinção entre terminais e subestruturas *superscript* e *subscript* [50].

Figura 3.7: Tipos de relações entre símbolos [50].

$ax + b$

Figura 3.8: Uma expressão com diferentes interpretações: $ax + b$, $aX + b$, $a^x + b$, $axtb$, etc. [27].

A escrita manuscrita irregular também dificulta a distinção entre as relações dos símbolos. Geralmente, essa irregularidade é causada por usuários inexperientes que apresentam variações na localização e alinhamento dos símbolos.

Vuong et. al [50] listam algumas técnicas para a análise estrutural: abordagens baseadas em gramática [48], transformação de árvore [55], Modelos Ocultos de Markov e árvore geradora mínima [44]. De acordo com [50], todas essas técnicas compartilham uma característica em comum: necessitam que todos os símbolos da expressões sejam preliminarmente reconhecidos.

3.2.4 Avaliação

Além dos problemas associados ao processo de reconhecimento das expressões, a avaliação das técnicas de reconhecimento constitui outro desafio nessa área. A avaliação de desempenho permite mensurar os progressos obtidos no desenvolvimento das técnicas e auxilia no planejamento para os próximos passos. Além disso, pode indicar erros na técnica e apontar suas razões, sendo possível, então, aprimorar a técnica a partir dessas informações. Também servem como ferramenta para

garantir qualidade nos resultados e como fundamento para aprovação da comunidade científica.

Segundo Chan & Yeung [7], as formas de avaliações de desempenho de técnicas de reconhecimento de expressões matemáticas podem ser agrupadas em três grandes categorias:

1. realizar o teste do método sobre um conjunto de expressões e categorizar os resultados por expressão;
2. realizar o teste do método sobre um conjunto de expressões e observar somente o desempenho relativo a uma determinada métrica (por exemplo, somente reconhecimento de símbolos, somente reconhecimento estrutural, etc.);
3. realizar o teste em algumas expressões comuns, geralmente escritas por poucos escritores. Como resultado espera-se que todas as expressões sejam corretamente reconhecidas. O objetivo para este tipo de avaliação é apenas confirmar que o método funciona ao menos para as expressões mais comuns.

O desempenho de um sistema depende da complexidade do conjunto de dados escolhido [3]. Portanto, é importante que os conjuntos de dados sejam realísticos, refletindo os tipos de expressões e variações de escrita a serem encontradas no domínio de aplicação [28]. De acordo com [16], é impossível mensurar o desempenho real de uma técnica sem testar todas as expressões do domínio. Além disso, o conjunto de dados deve ser disponibilizado publicamente e todas entradas devem ser rotuladas, isto é, cada entrada deve ser rotulada com a saída esperada.

Para mensurar e quantificar as avaliações, métricas adequadas devem ser escolhidas. De acordo com [25], as métricas devem ter 5 características:

- Devem refletir o objetivo da técnica (por exemplo, a quantidade de símbolos reconhecidos em uma técnica para reconhecimento de símbolos matemáticos);
- Devem ser abrangentes, isto é, considerar todos os objetivos da técnica (alto índice de caracteres reconhecidos corretamente e baixo índice de falsos-positivos no caso de uma técnica para reconhecimento de símbolos matemáticos);
- Devem ser compatíveis com avaliações humanas (um sistema dito excelente deve ser representado pela métrica);
- Devem ser conhecidas e aceitas pela comunidade científica;
- Devem ser claras para permitir a reprodução da avaliação.

Awal et. al. [3] fazem um levantamento de métricas para avaliar as técnicas para reconhecimento de expressões matemáticas manuscritas. De acordo com o estudo, três métricas são as mais utilizadas para avaliação:

Segmentação de símbolos: considerar a quantidade de símbolos corretamente segmentados (um símbolo é corretamente segmentado se todos os seus traços foram agrupados corretamente), independente se foram ou não corretamente reconhecidos [2,38];

Reconhecimento de símbolos: considerar a quantidade de símbolos corretamente reconhecidos, ignorando a estrutura da expressão. Por exemplo, se a expressão $x^2 + i$ com 4 símbolos for reconhecida como $x2 + 1$, considera-se apenas um erro de reconhecimento, isto é, o símbolo i reconhecido como 1. Nesse caso, apesar do símbolo 2 ter sido estruturalmente reconhecido incorretamente, o erro não é considerado [8];

Reconhecimento estrutural: verificar o desempenho da técnica em reconhecer a estrutura de expressões de entrada. Esses métodos de avaliação apenas verificam se a análise estrutural foi realizada corretamente, ignorando a etapa para reconhecimento de símbolos [54].

Há outras métricas, listadas em [25], tais como:

Posicionamento estrutural de símbolos: quantidade de símbolos localizados corretamente nas estruturas da expressão;

Reconhecimento de expressões: quantidade de expressões reconhecidas corretamente em um conjunto de expressões.

Tempo de processamento: tempo necessário para localizar e reconhecer um símbolo;

Lapointe [25] relaciona algumas dificuldades referentes a avaliação de técnicas para reconhecimento de expressões matemáticas, entre eles:

Comparação entre diferentes sistemas: a inexistência de uma metodologia consolidada para a avaliação do desempenho de reconhecimento faz com que os autores elaborem seu próprio critério de avaliação e utilizem diferentes conjuntos de dados em seus experimentos. Como consequência, comparações entre diferentes abordagens praticamente não são encontradas na literatura.

Reprodutibilidade dos experimentos para avaliação de desempenho: a metodologia utilizada nos trabalhos não são, por vezes, detalhadas o suficiente para permitir que a mesma seja reproduzida.

Domínio dos símbolos matemáticos: os conjuntos de dados consideram diferentes subconjuntos do domínio das notações matemáticas existentes. Assim, algumas técnicas apresentam bons resultados em domínios pequenos, mas apresentam resultados ruins em domínios maiores.

A inexistência de bases públicas consagradas e expressivas de expressões matemáticas manuscritas *online* [2, 3, 23, 37] é atualmente apontado como aspecto crítico. Em geral, cada autor considera seu próprio conjunto de dados para seus experimentos e, em sua maioria, esses conjuntos restringem-se à escrita de um número limitado de pessoas, expressões e símbolos [2, 14, 23, 49, 51].

Para expressões matemáticas *online*, destacam-se dois conjuntos de dados disponibilizados mediante registro, porém ainda não são referências consagradas na literatura. O primeiro é o conjunto de dados HAMEX [36], composto de 4350 expressões matemáticas, disponível em <http://www.hamex.org>

[//www.projet-depart.org/](http://www.projet-depart.org/). O segundo conjunto de dados é o ISI Nantes KAIST Mathematical Expression. Esse conjunto é composto por 1433 expressões matemáticas formatadas em InkML e trata-se de uma junção de conjuntos de dados de seis trabalhos diferentes, entre eles [2,22,36,38,39]. Esse conjunto está disponível em <http://www.irccyn.ec-nantes.fr/spip.php?article785>.

3.3 Abordagem proposta

Para gerar conjuntos de símbolos matemáticos manuscritos para treinamento e avaliação de classificadores de símbolos, uma maneira simples para coletar e rotular tais símbolos seria fazer com que as pessoas escrevessem individualmente os símbolos solicitados repetidas vezes. Em tal abordagem, uma vez que a identidade do símbolo é conhecida, sua rotulação seria automática. Haveria apenas a necessidade de verificar se o símbolo escrito é, de fato, o solicitado. Todavia, tanto a escrita como a verificação repetitiva de uma grande quantidade de amostras pode ser demorada, tediosa e cansativa para as pessoas.

Conforme mencionado na introdução, motivados pela necessidade de criação de conjuntos de símbolos e pelos fatos acima expostos, uma forma alternativa para a coleta de símbolos matemáticos manuscritos seria solicitar aos usuários a transcrição de expressões completas em vez da escrita individual de símbolos. As amostras dos símbolos seriam posteriormente extraídas dessas expressões.

Para justificar essa forma alternativa de coleta, os seguintes aspectos podem ser destacados:

- a escrita de expressões matemáticas completas é mais natural do que a escrita individual e repetida de símbolos. Além disso, os símbolos assim escritos são mais realísticos, ou seja, refletem melhor a forma na qual eles serão escritos dentro de expressões do que aqueles que são escritos de forma isolada;
- uma vez que a expressão matemática pode conter diversos símbolos, várias amostras de símbolos podem ser obtidas de uma única expressão;
- repetidas amostras de um mesmo símbolo podem ser obtidas repetindo-se o símbolo em uma expressão ou mesmo em várias expressões;
- uma vez que as expressões a serem transcritas podem ser diversificadas, a escrita deixa de ser uma tarefa repetitiva.

No entanto, em uma abordagem dessas torna-se necessária a segmentação e anotação de cada um dos símbolos presentes nas expressões. Para tratar essa necessidade, o método proposto neste trabalho consiste dos seguintes passos:

1. criação de um conjunto de expressões matemáticas manuscritas, com os símbolos devidamente segmentados e rotulados, denominadas de expressões modelo;

2. coleta de transcrições das expressões modelo, feitas por usuários, denominadas de expressões entrada;
3. associação de cada símbolo da expressão de entrada ao correspondente símbolo na expressão modelo, processo este a ser chamado de casamento de expressões;
4. transferência do rótulo dos símbolos da expressão modelo para os correspondentes símbolos na expressão entrada.

Convém ressaltar dois aspectos interessantes do método proposto. Primeiro, uma vez que as expressões modelo são criadas manualmente, nenhuma restrição quanto ao tipo de expressões nem de símbolos é imposta, e portanto o método é flexível quanto aos tipos de símbolos, estruturas de expressões e notações possíveis. Segundo, resultam do processo não apenas amostras de símbolos rotulados, mas também amostras de expressões completas e de subexpressões que podem ser utilizadas no desenvolvimento e avaliação de técnicas de análise estrutural.

Uma solução para o problema de casamento de expressões é descrita em detalhes no próximo capítulo. Na formulação apresentada, considera-se que ambas as expressões contém o mesmo número de símbolos e que estes encontram-se corretamente segmentados.

Para garantir que as expressões modelo e entrada tenham os símbolos corretamente segmentados e, também, que a quantidade de símbolos é exatamente a mesma em ambas, a coleta de dados é realizada por um software (*ExpressMatch* [1]) que induz essas duas propriedades. Mais especificamente, essas propriedades são implementadas da seguinte forma:

1. Segmentação de símbolos

Os símbolos são segmentados de acordo com o intervalo de tempo entre os traços. Dois traços subsequentes cujo intervalo de tempo entre o término do primeiro e o início do segundo é menor que um determinado limiar são considerados partes de um mesmo símbolo; caso contrário, são considerados partes de símbolos distintos. A medida que os traços são escritos, um *bounding box* é desenhado em tempo real ao redor dos traços que são considerados um símbolo, conforme visto na Figura 3.9. Assim, enquanto escreve, o usuário pode verificar visualmente se os símbolos estão sendo segmentados corretamente.

Figura 3.9: *ExpressMatch*: *bounding box* ao redor dos símbolos durante a escrita [1].

Essa abordagem depende de uma definição do que é considerado símbolo, uma vez que alguns símbolos são composições de outros. Na maioria dos casos, o senso comum é suficiente para caracterizar a noção do que é um símbolo, mas em alguns casos como nomes de funções, tais como $\cos$, $\sin$, $\log$, e operadores compostos, tais como $\pm$, $\dots$ e f' , não é claro quando deve-se considerar como um único símbolo.

Uma diretriz para se caracterizar um único símbolo é a seguinte: um conjunto de símbolos S é considerado como um único símbolo se, e somente se, sua representação é modificada caso seus componentes sejam considerados de forma isolada. Por exemplo, em “...” cada um dos pontos (.) são símbolos, porém, dentro do contexto de “...”, cada ponto não tem o mesmo significado de (.) quando considerado isoladamente.

No caso de “log”, cada um de seus componentes “l”, “o” e “g”, tem os significados respectivos de “l”, “o” e “g” tanto dentro de “log” quanto isoladamente. Assim, “log” não é considerado como um único símbolo e deve ser escrito como três símbolos sequenciais. O mesmo vale para $\ln$, $\lim$, $\sin$, $\cos$, etc. Em outros casos, como em f' , o significado de f dentro de f' não se altera e os dois símbolos devem ser escritos separadamente.

2. Quantidade de símbolos

Uma vez que a quantidade de símbolos na expressão modelo é conhecida e a segmentação ocorre simultaneamente à transcrição, é simples verificar se a quantidade de símbolos é a mesma em ambas após o término da transcrição. Caso não seja, o usuário pode ser alertado. Desta forma o usuário poderá detectar símbolos faltantes, excedentes ou ainda segmentados incorretamente. Para facilitar a correção de eventuais erros, o software permite ao usuário apagar um dado símbolo e reescrevê-lo. É importante observar que o fato do número de símbolos coincidir não garante que a transcrição foi realizada corretamente. Uma etapa de verificação manual das transcrições é inevitável para garantir uma base de dados sem erros.

3.4 Trabalhos relacionados

Dentre alguns trabalhos na literatura que buscam formas para facilitar a rotulação de símbolos matemáticos *online*, podemos citar os seguintes.

Li et. al. [26] propõem um sistema para criação de conjuntos de símbolos matemáticos rotulados a partir de expressões matemáticas completas. Apesar da rotulação de símbolos ser realizada manualmente pelo usuário, o objetivo da abordagem é minimizar sua interação no sistema, de modo que a expressão inteira seja rotulada com a menor quantidade de rótulos informados pelo usuário.

A abordagem proposta é apresentada na Figura 3.10. Dada uma expressão matemática já segmentada, a abordagem agrupa os segmentos semelhantes (considera-se que um segmento pode conter traços de mais um símbolo). O agrupamento é realizado com a utilização de um algoritmo de agrupamento hierárquico que considera uma métrica de distância para os segmentos semelhantes. O

ideal é que os grupos sejam formados por segmentos que tenham a menor distância entre si e maior distância com elementos de outros agrupamentos.

Após a etapa de agrupamento, um elemento de cada grupo é escolhido como representante. Solicita-se ao usuário que rotule manualmente cada traço do elemento representante e o algoritmo encarrega-se de propagar o rótulo informado para outros segmentos do agrupamento. Durante a propagação, os traços mais semelhantes ao traço rotulado são os que possuem menor distância.

O esforço do usuário é reduzido, uma vez que não é necessária a rotulação individual de símbolos da expressão. Os autores afirmam que a técnica é efetiva para casos de símbolos de expressões matemáticas, porém são muito dependentes da segmentação inicial fornecida.

Por utilizar técnicas de agrupamento, observa-se uma limitação dessa técnica para os casos de expressões matemáticas com poucas repetições de símbolos. Devido a existência de muitos agrupamentos diferentes, o usuário deverá rotular manualmente praticamente todos os símbolos da expressão.

Figura 3.10: Processo de rotulação de símbolos baseado em agrupamentos de segmentos [26].

Sminova & Watt [42] propõem uma ferramenta (denominada *Mathink*) para rotulação de símbolos matemáticos com o objetivo de gerar uma grande quantidade de amostras para treinamento,

ajustes de classificadores de símbolos e realização de experimentos. No processo de aquisição de amostras, primeiramente expressões matemáticas pré-definidas (modelos) são rotuladas manualmente e carregadas na ferramenta. A ferramenta, então, apresenta individualmente as expressões ao usuário, que realiza a transcrição na própria ferramenta.

Após a transcrição, um classificador é aplicado em cada símbolo da expressão transcrita. A validação dos rótulos informados pelo classificador é realizada por uma pessoa avaliadora que corrige manualmente os rótulos incorretos dos símbolos. Os autores indicam que esse método é válido para símbolos facilmente reconhecíveis, isto é, com altos índices de classificações corretas. Quando exige-se muitas correções manuais na expressão, os autores propõem a utilização de um outro método, denominado de *Annotation Copying*.

O método *Annotation Copying* considera a sequência de escrita das expressões modelos. Basicamente, o rótulo do primeiro símbolo escrito na expressão modelo é associado para o primeiro símbolo escrito na expressão transcrita (não rotulada), o segundo da expressão pré-definida para o segundo da expressão transcrita, e assim por diante. Esse processo não pode ser feito de maneira totalmente automática, pois existe a possibilidade de que os símbolos estejam escritos em diferentes ordens, como é o caso, por exemplo, de frações, parênteses e raízes.

Apesar das restrições apontadas para a funcionalidade de aquisição de amostras, a ferramenta *Mathink* se mostra bastante completa pois considera diversas técnicas para segmentação, pré-processamento, reconhecimento de símbolos, predição de símbolos e análise estrutural das expressões.

McLean et. al. [27] propõem a criação de um corpus de expressões matemáticas completas com símbolos rotulados. Em seu trabalho, as expressões correspondem a produções de uma gramática *fuzzy* relacional. Essa gramática busca capturar as possíveis relações espaciais entre diferentes símbolos do alfabeto considerado (neste caso, o alfabeto é composto por expressões matemáticas utilizadas até o início do nível universitário). Algumas estratégias são utilizadas para controlar o tamanho das expressões, garantir variedade de expressões e variedade de posições relativas entre símbolos.

Uma produção da gramática é representada por uma *string* que codifica uma expressão. Essas *strings* são convertidas para a notação $\text{\LaTeX}$ e em seguida são geradas as imagens das expressões correspondentes. Voluntários são, então, orientados a transcrever essas expressões. Para rotular os símbolos da expressão manuscrita, os autores procuram associar grupos de traços aos símbolos terminais na *string* original que descreve a expressão. Para isso são utilizados classificadores que associam *scores* a conjuntos de traços e também classificadores de relação que associam *scores* às posições relativas entre dois grupos de traços.

Os autores afirmam que, se os traços forem agrupados corretamente, a tendência é que a rotulação dos símbolos seja realizada corretamente. Porém, um problema apontado na metodologia é a alta taxa de rejeição de conjuntos de traços, quando os mesmos não podem ser casados com símbolos terminais da *string* em questão.

Um aspecto interessante dessa metodologia é que ela trata o problema de definir os tipos de expressões que farão parte do corpus e também o problema de rotular automaticamente as

respectivas expressões manuscritas. O corpus gerado pela abordagem proposta encontra-se em <http://www.cs.uwaterloo.ca/scg/mathbrush/mathdata.shtml>. De fato, estabelecer uma gramática cujas produções reflitam fielmente a distribuição de expressões de um dado contexto de aplicação não é simples.

Capítulo 4

Casamento de expressões matemáticas

Neste capítulo é descrito o método proposto para o casamento de expressões matemáticas. Neste método, supõe-se que uma das expressões é uma expressão modelo, com os símbolos individualmente segmentados e identificados, e a outra é uma transcrição da expressão modelo (denominada de expressão de entrada), com os símbolos também individualmente segmentados, porém não identificados. O objetivo do casamento de expressões matemáticas é associar cada símbolo da expressão transcrita ao correspondente símbolo na expressão modelo.

Conforme mencionado na introdução, esse problema é formulado como um problema de associação simples. Neste capítulo, primeiramente é descrita uma visão geral dessa formulação e, em seguida, os detalhes de cada uma das principais etapas da mesma. Por fim, uma proposta para a avaliação de casamentos de expressões é apresentada.

4.1 Formulação do problema de associação

O casamento de expressões pode ser visto como um problema de associação simples, pois a associação dos símbolos ocorre em uma situação em que ambas as expressões tem a mesma quantidade de símbolos.

Em um problema de associação, custos de associação entre os pares devem ser definidos. Os custos ideais são aqueles que associam custo nulo para o par correto e custo alto para os pares que formam associações incorretas. No contexto de expressões matemáticas, uma vez que as mesmas são arranjos bidimensionais, a informação estrutural da expressão (posição relativa entre os símbolos na expressão) e características do símbolo (forma e traços) são informações relevantes para estabelecer esse custo.

Nesse caso, o custo de deformação estrutural descrito na seção 2.4, que considera uma combinação de informações referentes à forma (atributos locais) e de aspectos estruturais (atributos globais), parece adequado. Este custo pode ser adaptado ao problema de casamento de expressões matemáticas: os atributos globais referem-se a aspectos da expressão como um todo, como a disposição dos símbolos dentro da expressão, e os atributos locais referem-se aos aspectos associados aos símbolos

matemáticos, como a altura, largura, proporção, quantidade de traços, o *shape context*, relação de vizinhança local, entre outros.

A representação de expressões por ARGs surge naturalmente nesse contexto pois as arestas podem armazenar informações estruturais da expressão e vértices podem armazenar informações referentes aos símbolos. A expressão modelo é, então, representada por um grafo, denominado de grafo modelo e a expressão de entrada é representada por outro grafo, denominado de grafo de entrada.

Deste modo, o problema de associação de símbolos citado anteriormente pode ser entendido como um problema de associação de vértices do grafo de entrada para vértices do grafo modelo. Logo, o problema de associação pode ser visto como o problema de emparelhamento de grafos bipartidos, como visto na seção 2.2.

O custo de deformação, conforme definido, pressupõe grafos espacialmente alinhados, porém as expressões podem estar espacialmente em posições distintas e apresentarem escalas diferentes. Por isso, logo após a construção dos grafos a partir das expressões, é importante o alinhamento do grafo de entrada sobre o grafo modelo, tornando os vértices correspondentes espacialmente mais próximos.

Resumidamente, a técnica para casamento de expressões matemáticas utilizada neste trabalho consiste em quatro etapas principais:

- (i) construção do grafo modelo e de entrada correspondentes às expressões modelo e de entrada respectivamente (a ser apresentado na seção 4.2);
- (ii) normalização do grafo de entrada em relação ao grafo modelo, com respeito à posição espacial e à escala (a ser apresentado na seção 4.3);
- (iii) cálculo do custo de associação de cada vértice do grafo de entrada, em relação a cada vértice do grafo modelo (a ser apresentado na seção 4.4);
- (iv) construção do grafo bipartido balanceado a partir dos custos de associação (como visto na seção 2.2.1).
- (v) cálculo da associação um-para-um de custo mínimo do grafo bipartido balanceado, solucionado usando o Algoritmo Húngaro (como visto na seção 2.2.2).

4.2 Representação de expressões matemáticas por grafos

Conforme citado na seção anterior, as expressões matemáticas são arranjos bidimensionais de símbolos. Assim, sua informação estrutural pode ser representada por ARGs, nos quais os vértices correspondem aos símbolos e as arestas correspondem às relações espaciais entre eles. Mais especificamente, as coordenadas (x, y) do centróide do símbolo da expressão são consideradas como um atributo do vértice correspondente no ARG. Informações adicionais, como o *shape context* dos respectivos símbolos, podem ser atribuídas aos vértices.

As relações espaciais entre os símbolos dentro da expressão são representadas por arestas que conectam os respectivos vértices no ARG. Tais arestas armazenam informações quanto à distância e orientação entre os vértices. As relações espaciais podem ser, por exemplo, estabelecidas pela Triangulação de Delaunay (seção 2.3), pelo grafo completo ou por outro critério mais adequado, como os k vértices mais próximos, apenas os vértices a uma certa distância d um do outro, entre outros. Todos esses critérios podem ser testados experimentalmente para que seja escolhido o mais apropriado para cada situação.

Assim como as expressões, no espaço bidimensional os ARGs são envolvidos por um *bounding box*, que contém todos os vértices, e possuem centróide, descrito na seção 3.1.

4.3 Normalização espacial e de escala

Dados dois grafos estruturalmente similares, um deles denominado de grafo modelo e o outro de grafo de entrada, o objetivo da normalização é alinhar o grafo de entrada com base no grafo modelo tanto espacialmente quanto em escala. Esse processo é composto pelas seguintes etapas:

1. Obtém-se o centróide do grafo modelo a partir de seu *bounding box*. O mesmo é realizado com o grafo de entrada.
2. O grafo de entrada (e conseqüentemente seus vértices) é transladado de forma que as coordenadas de seu centróide coincidam com as coordenadas do centróide do grafo modelo.
3. A largura e altura do *bounding box* do grafo de entrada são independentemente ajustadas para igualarem-se com a largura e altura do grafo modelo, respectivamente. Nestas operações, as coordenadas de cada vértice do grafo de entrada também são proporcionalmente ajustadas.

Para ilustrar o efeito da normalização, a Figura 4.1 apresenta um exemplo no qual no lugar dos vértices são mostrados os respectivos símbolos. Neste caso, apenas as coordenadas espaciais dos vértices são alteradas; os tamanhos dos símbolos associados não são modificados [18].

$$\begin{aligned}
& c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \\
& \text{(a) Entrada} \\
& c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \\
& \text{(b) Modelo} \\
& c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \\
& \text{(c) Entrada normalizada}
\end{aligned}$$

Figura 4.1: Normalização do grafo de entrada com base no grafo modelo. Para efeitos de ilustração, os vértices foram representados pelos respectivos símbolos da expressão associada.

4.4 Custos de associação

Se um vértice u_i no grafo de entrada corresponde a um dado vértice v_j do grafo modelo, u_i tende a estar aproximadamente na mesma posição espacial relativa a de v_j dentro do grafo. Portanto, se trocarmos o vértice v_j por u_i , é razoável supor que não haverá mudanças significativas no grafo modelo.

O custo de deformação, apresentado na seção 2.4, possibilita capturar essa informação de modo que a deformação do grafo modelo seja minimizada de um modo geral. Logo, o custo definido pela Equação 2.2 parece ser uma escolha adequada para caracterizar a qualidade de uma associação:

$$c(u_i, v_m) = \alpha c_v(u_i, v_m) + (1 - \alpha) c_e(u_i, v_m).$$

O referido custo integra um termo relacionado aos atributos de vértice e outro relacionado aos atributos das arestas. No caso de vértices, neste trabalho é proposto o uso de *shape context* (seção 2.5). Para o cálculo do *shape context* são considerados dois conjuntos de pontos distintos.

Um deles corresponde a n pontos equidistantes sobre os traços do próprio símbolo. Neste caso, o *shape context* é calculado para cada ponto do símbolo e, para cada ponto, a contagem do histograma é realizada sobre os $n - 1$ pontos restantes, conforme apresentado na Figura 4.2.

Por vezes os traços de símbolos podem apresentar certas irregularidades, como pontos redun-

Figura 4.2: *Shape context* de um ponto em relação aos outros pontos do símbolo.

dantes e espaçamentos irregulares entre os pontos. Neste trabalho, considera-se a técnica de pré-processamento apresentada em [19]. Nessa técnica, primeiramente são removidos os pontos duplicados e os “ganchos” das extremidades dos traços. Os ruídos são filtrados com o uso de uma técnica de alisamento (*smoothing*) e, então, normalizados. As distâncias entre os pontos dos traços são equalizadas. De acordo com [19], essa técnica aprimora o desempenho de reconhecimento em 10%. A Figura 4.3 apresenta a aplicação da técnica em uma expressão com 4 símbolos.

(a) Expressão escrita pelo usuário: apenas alguns pontos dos traços são capturados pela ferramenta. (b) Aplicação do pré-processamento na expressão: pontos equidistantes nos traços dos símbolos.

Figura 4.3: Pré-processamento para a expressão $n! - 1$.

Outro conjunto considera os centróides de cada símbolo da expressão. O *shape context* é, então, calculado para cada centróide em relação aos demais centróides da expressão, conforme apresentado na Figura 4.4.

Enquanto a primeira informação tem uma característica local, isto é, relacionada à forma do símbolo (*shape context* local), a segunda tem uma característica global, relacionando-se com a posição relativa do símbolo aos demais símbolos na expressão (*shape context* global). Assim, pode-se definir o custo $c_{v(local)}$ e o custo $c_{v(global)}$.

O custo local $c_{v(local)}$ deve capturar a dissimilaridade entre dois símbolos de expressões distintas. Considerando n pontos sobre os traços de cada símbolo, a intenção é associar cada ponto do primeiro símbolo com um ponto do segundo símbolo. Como cada ponto possui um *shape context*, é possível,

Figura 4.4: *Shape context* de um símbolo em relação a outros símbolos da expressão.

então, utilizá-los para mensurar a similaridade entre pontos. Logo, para cada par de pontos, haverá uma medida de similaridade, definida pela Equação 2.4.

Conforme apresentado na seção 2.4, a melhor associação entre dois conjuntos de pontos é a minimização dessa medida como um todo, considerando todas as associações entre pares de pontos. Como uma quantidade igual de pontos em cada símbolo está sendo considerada, então é possível utilizar o Algoritmo Húngaro, apresentado na seção 2.2.2, para definir a associação de pontos de custo mínimo.

Assim, o custo local $c_{v(local)}$ é a média aritmética calculada a partir da associação ótima entre os conjuntos de pontos, isto é,

$$c_{v(local)} = \frac{1}{n} H(\pi) \quad (4.1)$$

na qual n é a quantidade de pontos de um símbolo (e, também, a quantidade de mapeamentos entre pontos) e π é a associação que minimiza a medida de similaridade H , definida pela Equação 2.5.

Dado um par de vértices (u_i, v_j) , sendo u_i do grafo de entrada e v_j do grafo modelo, o custo global $c_{v(global)}$ entre dois vértices é definido por uma equação muito semelhante à Equação 2.4:

$$c_{v(global)}(u_i, v_j) \equiv C(u_i, v_j) = \frac{1}{2} \sum_{k=1}^K \frac{[hu_i(k) - hv_j(k)]^2}{hu_i(k) + hv_j(k)} \quad (4.2)$$

na qual K é a quantidade total de *bins* do *shape context* e hu_i e hv_j são, respectivamente, os histogramas associados a u_i e v_j . Caso $hu_i(k) = hv_j(k) = 0$, o correspondente termo não é considerado no somatório.

O intuito é que $c_{v(global)}$ seja alto caso os dois vértices estejam em posições muito diferentes dentro das respectivas expressões e baixo quando os mesmos estejam em posições relativamente parecidas.

O custo de vértice c_v (Equação 2.2) é, então, definido pela seguinte equação:

$$c_v = \gamma c_{v(local)} + (1 - \gamma) c_{v(global)} \quad (4.3)$$

na qual $c_{v(local)}$ representa o custo associado às características locais dos símbolos, $c_{v(global)}$ representa o custo associado às características globais, e γ é um número real entre 0 e 1, usado para ponderar a importância entre os dois termos.

Para os custos de aresta, utilizamos o custo definido em [33], definido da seguinte maneira:

$$c_e(u_i, v_j) = \frac{1}{\#E(v_j)} \sum_{e \in E(v_j)} c_{vec}(\mathbf{e}, \tilde{\mathbf{e}}) \quad (4.4)$$

na qual $E(v_j)$ é o conjunto de arestas incidentes em v_j e $\#E(v_j)$ representa sua cardinalidade, $\mathbf{e}$ é o vetor correspondente à aresta e , e $\tilde{\mathbf{e}}$ é o vetor obtido substituindo-se a extremidade v_j de $\mathbf{e}$ por u_i .

O custo $c_{vec}(\mathbf{e}, \tilde{\mathbf{e}})$ entre dois vetores que incidem em um mesmo vértice é dado por:

$$c_{vec}(\mathbf{e}, \tilde{\mathbf{e}}) = \beta \frac{|\cos\theta - 1|}{2} + (1 - \beta) \frac{||\mathbf{e}| - |\tilde{\mathbf{e}}||}{C} \quad (4.5)$$

onde θ define o ângulo entre os dois vetores $\mathbf{e}$ e $\tilde{\mathbf{e}}$, $|\cdot|$ denota o comprimento do vetor, β é um número real entre 0 e 1 e C é uma constante de normalização, representado pelo tamanho da diagonal do *bounding box* do grafo modelo. Uma vez que o custo de aresta definido considera apenas a deformação no grafo modelo, o grafo de entrada é construído sem arestas, isto é, representado apenas por um conjunto de vértices.

4.5 Avaliação das associações

Várias expressões modelo podem ser consideradas para o casamento de expressões. Para cada expressão modelo podem existir várias instâncias de expressões de entrada. Uma vez que não existe nenhuma informação sobre a construção de bons modelos, para efeitos de avaliação da técnica de casamento proposta neste trabalho, uma expressão modelo e suas respectivas transcrições são consideradas como uma classe de expressões.

Com a finalidade de avaliar a qualidade do casamento de expressões matemáticas obtido (e, conseqüentemente, da técnica proposta), um esquema de casamento cruzado é proposto: para cada classe, cada expressão é utilizada como expressão modelo com a qual as demais expressões da classe são casadas, isto é, todas as expressões são utilizadas como modelo uma vez.

Seja m a quantidade de classes existentes no conjunto. Dada uma classe c , n_c é a quantidade de instâncias nela e k_c é a quantidade de símbolos em cada instância. Definimos:

- $t(c)_{ij}$ o número de símbolos associados corretamente quando se considera o casamento entre a expressão modelo E_i e a expressão de entrada E_j , ambas pertencentes a classe c . Logo, $t(c)_{ij}$ é um número entre 0 e k_c .
- $T(c)_i = \sum_{\substack{j=1 \\ j \neq i}}^{n_c} t(c)_{ij}$ é o número total de símbolos associados corretamente, considerando todos os

casamentos quando a expressão modelo é fixada em E_i . $T(c)_i$ é um número entre 0 e $k_c(n_c - 1)$;

- $t(c) = \sum_{i=1}^{n_c} T(c)_i = \sum_{i=1}^{n_c} \sum_{\substack{j=1 \\ j \neq i}}^{n_c} t(c)_{ij}$ é o número total de símbolos associados corretamente, considerando o conjunto de todos os $n_c(n_c - 1)$ casamentos. $t(c)$ é um número entre 0 e $k_c n_c(n_c - 1)$.

A partir dos termos acima, temos:

- **associação correta média por casamento:** é definida como

$$\overline{t(c)} = \frac{t(c)}{n_c(n_c - 1)} = \frac{\sum_{i=1}^{n_c} \sum_{\substack{j=1 \\ j \neq i}}^{n_c} t(c)_{ij}}{n_c(n_c - 1)}, \quad (4.6)$$

que é um número entre 0 e k_c . Esse valor indica a quantidade média de associações corretas (acertos) de uma expressão modelo dentro de uma classe c , isto é, considerando todas as expressões contidas em c no esquema de casamento cruzado. Seu desvio padrão é definido

como $\sqrt{\frac{\sum_i^{n_c} \sum_{\substack{j=1 \\ j \neq i}}^{n_c} (\overline{t(c)} - t(c)_{ij})^2}{n_c(n_c - 1)}}$.

- **associação correta média por casamento normalizada:** é definida como

$$\overline{s(c)} = \frac{\overline{t(c)}}{k_c}, \quad (4.7)$$

que é um número entre 0 e 1. Esse valor indica a porcentagem média de associações corretas de uma expressão modelo dentro de uma classe c . Seu desvio padrão é definido como

$$\sqrt{\frac{\sum_i^{n_c} \sum_{\substack{j=1 \\ j \neq i}}^{n_c} (\overline{s(c)} - t(c)_{ij}/k_c)^2}{n_c(n_c - 1)}}.$$

- **associação correta média global por casamento:** é definida como:

$$\overline{S} = \frac{\sum_{c=1}^m \overline{s(c)}}{m} = \frac{\sum_{c=1}^m (\overline{t(c)}/k_c)}{m}, \quad (4.8)$$

no qual $\overline{S}$ é um valor entre 0 e 1. Esse valor indica a porcentagem média de acertos de uma expressão modelo considerando todas as classes do conjunto de dados. Seu desvio padrão é

definido como $\sqrt{\frac{\sum_{c=1}^m (\overline{S} - \overline{s(c)})^2}{m}}$.

Na prática, para apresentação dos resultados, os índices que nos interessam são: **associação correta média por casamento**, **associação correta média por casamento normalizada** e a **associação correta média global por casamento**.

Capítulo 5

Resultados Experimentais

Resultados preliminares sobre um conjunto de dados, denominado conjunto A, foram publicados no artigo *Automatic Labeling of Handwritten Mathematical Symbols via Expression Matching* [18], apresentado no *8th IAPR - TC-15 Workshop on Graph-based Representations in Pattern Recognition*.

Posteriormente, o método foi melhorado (tal qual é descrito neste texto) e aplicado sobre um outro conjunto de dados, denominado conjunto B, bem maior que o primeiro. Neste capítulo, descreve-se inicialmente as características desses conjuntos de dados e, em seguida, apresenta-se os desempenhos do método para diferentes valores de parâmetros, avaliados sobre o conjunto B. Por fim, os valores de parâmetros dos melhores resultados obtidos são usados para processar o conjunto A, com o intuito de validar a abordagem proposta.

5.1 Descrição dos conjuntos de dados

5.1.1 Conjunto A

Na composição do conjunto de dados A, oito classes de expressões (ilustradas na Figura 5.1) foram tipografadas em L^AT_EX e impressas em papel, e oito voluntários foram instruídos a escreverem essas expressões no *tablet* de mesa WACOM Intuos 3 Graphics PTZ-930.

Logo, o número de classes (tipos de expressões) é $m = 8$ e o número de instâncias por classe é $n = 8$, resultando em um total de 64 amostras de expressões. A Figura 5.2 apresenta a quantidade de símbolos e instâncias por classe de expressão.

Para a coleta desses dados, utilizou-se uma adaptação da ferramenta *ExpressMath* [41]. Nessa adaptação, não há uma funcionalidade para verificar se a quantidade de símbolos escritos na transcrição é igual à quantidade de símbolos da expressão modelo. Deste modo, houveram transcrições com símbolos faltantes e símbolos incorretamente segmentados.

Essa versão da ferramenta também não permite a rotulação de símbolos das expressões modelo

$$ax^2 + bx + c = 0$$

(a) Classe 1

$$\sqrt{\alpha\beta\delta} \geq \frac{1}{\varepsilon\delta}$$

(b) Classe 2

$$C_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

(c) Classe 3

$$M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

(d) Classe 5

$$P_a = \frac{1}{\sqrt{2\pi}} \int_a^{\infty} e^{-\frac{t^2}{2}} dt$$

(e) Classe 6

$$\Phi[n-k] = \begin{cases} 1, & n=k \\ 0, & n \neq k \end{cases}$$

(f) Classe 7

$$p(w_i|x) = \frac{p(x/w_i)P(w_i)}{\sum_{j=1}^c p(x/w_j)P(w_j)}$$

(g) Classe 4

$$\sum_{i=1}^n i^p = \frac{1}{p+1} \left[(n+1)^{p+1} - 1 - \sum_{i=1}^p \sum_{j=0}^{p-1} \binom{p+1}{j} i^j \right]$$

(h) Classe 8

Figura 5.1: Exemplos das expressões do conjunto A.

Figura 5.2: Quantidade de símbolos e instâncias por classe de expressão do conjunto A.

e, portanto, todos os símbolos foram rotulados manualmente. As irregularidades na segmentação também foram corrigidas manualmente. Os símbolos que foram incorretamente segmentados foram corrigidos e os símbolos excedentes foram excluídos manualmente.

5.1.2 Conjunto B

Para a coleta do conjunto de dados B, utilizou-se o programa *ExpressMatch* [1] em um *tablet PC* HP Pavilion tx2500, com 12.1 polegadas e uma resolução de 1280×800 pixels.

A ferramenta *ExpressMatch* permite, por meio de uma interface simples e intuitiva, a entrada de expressões modelo e a rotulação de seus símbolos. Dado um conjunto de expressões modelo, a ferramenta exibe aleatoriamente diferentes expressões do conjunto para que os usuários façam sua transcrição. Essas transcrições, instâncias das respectivas expressões modelo, são armazenadas como expressões de entrada. Além disso, a ferramenta permite a inserção de mais de uma instância por modelo, possibilita o gerenciamento de quais expressões modelo foram transcritas e a quantidade de instâncias criadas para cada modelo por usuário, sendo possível ao usuário continuar a transcrição em diferentes momentos.

O software realiza a segmentação dos símbolos a partir da informação temporal entre a escrita de dois símbolos, como visto na seção 3.3. Caso traços de símbolos diferentes sejam agrupados, o usuário pode utilizar uma função do software para refazer o último símbolo escrito. A Figura 5.3 apresenta a tela do software para coleta das transcrições do usuário.

Figura 5.3: Tela para coleta de expressões no software *ExpressMatch* [1].

As expressões modelo correspondem a 56 expressões matemáticas obtidas dos Capítulos 2 e 4 de [57] sem um critério específico (exemplos dessas expressões são apresentadas no apêndice A). Solicitou-se a cada usuário que transcrevesse uma instância da maior quantidade possível dessas expressões de acordo com sua disponibilidade de tempo. Uma quantidade de 926 transcrições foram

obtidas. A Figura 5.4 apresenta a quantidade de símbolos e instâncias por classe de expressão.

Figura 5.4: Quantidade de símbolos e instâncias por classe de expressão do conjunto B.

Após a coleta, uma verificação visual foi realizada nas expressões do conjunto como forma de avaliar a qualidade das transcrições. Três tipos de erros foram encontrados: erros de segmentação, erros de semântica (devido ao posicionamento incorreto de símbolos ou símbolos incorretos) e símbolos ilegíveis.

A Figura 5.5(a) apresenta um erro de segmentação onde dois símbolos consecutivos (“!” e “–”) foram incorretamente segmentados. As Figuras 5.5(b) e 5.5(c) apresentam erros causados por símbolos incorretos na expressão. A Figura 5.5(d) apresenta uma transcrição com um símbolo extra (ponto) entre “a” e “s”. A Tabela 5.1 apresenta a quantidade de transcrições por tipo de erro.

Tabela 5.1: Transcrições desconsideradas do Conjunto B

Tipo de erro	Total
Segmentação incorreta	8
Semântica (posicionamento incorreto)	5
Semântica (símbolos incorretos)	4
Símbolos ilegíveis	8
Total	25

Assim, de um total de 926 transcrições, 25 foram desconsideradas. As 901 transcrições restantes foram somadas às 56 expressões modelo para a avaliação da técnica. No total, são 81 tipos de símbolos diferentes. O número médio de símbolos por expressão é 22.5, sendo o mínimo 4 e o máximo 66.

Figura 5.5: Exemplos de expressões desconsideradas: (a) Incorreta segmentação da expressão $n! - 1$, onde o ponto, “.”, e o sinal de subtração, “-”, foram considerados como sendo traços de um mesmo símbolo (as linhas ao redor dos símbolos apresentam o *bounding-box* de cada símbolo); (b) O símbolo “+” foi escrito no lugar de “-” (a correta expressão deveria ser $(a \frac{1-t^2}{1+t^2}, \frac{2bt}{1+t^2})$); (c) A posição espacial de x indica a multiplicação $x\sqrt{\frac{a}{b}}$ e a correta expressão seria $\frac{x\sqrt{a}}{\sqrt{b}}$; (d) Um ponto extra entre os símbolos “a” e “s”.

5.2 Avaliação de parâmetros

Para os experimentos deste trabalho, as seguintes configurações foram utilizadas:

- No pré-processamento dos símbolos, visando pontos bem distribuídos, foram selecionados 30 pontos dispostos de forma equidistante sobre os traços do símbolo;
- Para o *shape context* local, foram escolhidas 4 regiões na distância e 10 regiões angulares, totalizando 40 *bins*, semelhante ao visto na Figura 4.2. A quantidade de *bins* foi escolhida a partir de resultados vistos em [35];
- Para o *shape context* global, foram escolhidas 2 regiões na distância e 4 regiões angulares, totalizando 8 *bins*, semelhante ao visto na Figura 4.4. A quantidade de *bins* foi escolhida empiricamente.

Os parâmetros avaliados durante os experimentos foram:

- β : pondera entre o coeficiente de tamanho e angular do custo de aresta (Equação 4.5). Quando $\beta = 0$, apenas o coeficiente de tamanho é considerado e quando $\beta = 1$ apenas o coeficiente angular é considerado;
- γ : pondera entre o *shape context* global e local do custo de vértice (Equação 4.3). Quando $\gamma = 0$, apenas o *shape context* global é considerado e quando $\gamma = 1$ apenas o *shape context* local é considerado;
- α : pondera entre o custo de aresta e de vértice (Equação 2.2). Quando $\alpha = 0$, apenas o custo de aresta é considerado e quando $\alpha = 1$ apenas o custo de vértice é considerado.

A fim de evitar a grande quantidade de possíveis combinações de valores dos parâmetros, primeiramente o método foi avaliado com diferentes valores para os custos de aresta (β) e de vértice (γ) de forma independente. Os valores de ambos que apresentaram melhor desempenho foram, então, fixados para avaliar o método com diferentes valores para α .

Para a análise dos resultados, foram utilizados valores entre 0.0 e 1.0 para os parâmetros α , β e γ . Além disso, para o custo de aresta, foram utilizados dois tipos de conectividade nas arestas do grafo modelo: Triangulação de Delaunay e grafo-completo. Os resultados desta seção referem-se aos obtidos sobre o conjunto B.

Custo de aresta

Os gráficos 5.6(a) e 5.6(b) apresentam os resultados do método obtidos utilizando-se apenas o custo de aresta (parâmetro β) e a Triangulação de Delaunay no grafo modelo. Nota-se que o desempenho da técnica é inversamente proporcional ao tamanho da expressão. O melhor resultado foi obtido com $\beta = 0.25$, apresentando uma associação correta média global por casamento de 89,9%. Se avaliarmos esse resultado pela quantidade de símbolos por expressão, verificamos que a associação correta média por casamento para expressões com até 23 símbolos é 96,32% e, considerando apenas expressões com mais de 23 símbolos, o desempenho médio decresce para 78,25%.

(a) Resultados com variações do custo de aresta. (b) Resultados normalizados com variações do custo de aresta.

Figura 5.6: Associação correta média por casamento obtida com Triangulação de Delaunay e $\beta = \{0.0, 0.25, 0.5, 0.75, 1\}$ para o Conjunto B.

Os resultados do método considerando a conectividade de grafo completo para o grafo modelo são apresentados no gráfico 5.7(a) e no gráfico 5.7(b). Observa-se que, neste caso, há mais variação entre os resultados obtidos com diferentes valores de β : ao longo do gráfico, nos casos em que as expressões possuem poucos símbolos, os resultados são relativamente próximos, porém, para as expressões com maior quantidade de símbolos, há mais discrepância entre os resultados. O melhor resultado foi obtido com $\beta = 1.0$, isto é, quando apenas o coeficiente angular é considerado, apresentando-se uma associação correta média global por casamento de 98,3%. O pior desempenho foi obtido com

$\beta = 0.0$, ou seja, quando apenas o coeficiente de tamanho é considerado, com uma associação correta média global de 83,5%.

(a) Resultados com variações do custo de aresta. (b) Resultados normalizados com variações do custo de aresta.

Figura 5.7: Associação correta média por casamento obtida com grafo completo e $\beta = \{0.0, 0.25, 0.5, 0.75, 1\}$ para o Conjunto B.

Comparando-se a associação correta média global por casamento obtida a partir do uso dos dois tipos de conectividade do grafo modelo, o melhor resultado foi obtido com o uso do grafo completo em todos os valores considerados para o parâmetro β . A diferença entre o melhor desempenho obtido com o grafo completo ($\beta = 1.0$) e o melhor desempenho obtido com a Triangulação de Delaunay ($\beta = 0.25$) é de 8,4%. Esses resultados comparativos são apresentados na Figura 5.8.

Figura 5.8: Associação correta média global por casamento utilizando as configurações de grafo completo e Triangulação de Delaunay para o grafo modelo.

O desempenho superior apresentado com a utilização do grafo completo pode ser explicado pela presença de mais conexões em cada vértice, o que afeta diretamente o custo de aresta (Equação 4.4).

Por considerar tanto pontos próximos quanto mais distantes, a deformação estrutural dos vértices do grafo completo considera mais referências do que os vértices de uma Triangulação de Delaunay. Este último considera somente pontos relativamente mais próximos e mostram-se mais sensíveis para modificações muito bruscas de posição. Assim, a estrutura do grafo completo mostra-se mais robusta para as deformações sofridas pelo grafo modelo.

Custo de vértice

No cálculo do custo de vértice, γ pondera entre considerar apenas o *shape context* global ($\gamma = 0.0$) e *shape context* local ($\gamma = 1.0$). Pelos gráficos 5.9(a) e 5.9(b), observa-se claramente que, quando emprega-se o parâmetro $\gamma = 1.0$, o índice de associação correta média diminui conforme são consideradas as expressões com maior quantidade de símbolos. Além disso, os resultados apresentados pela técnica utilizando-se $\gamma = 1.0$ são inferiores aos resultados obtidos com outros valores para γ .

(a) Resultados com variações do custo de vértice. (b) Resultados normalizados com variações do custo de vértice.

Figura 5.9: Associação correta média por casamento obtida com $\gamma = \{0.0, 0.25, 0.5, 0.75, 1\}$ para o Conjunto B.

A Figura 5.10 apresenta a associação correta média global por casamento obtida com variações de γ . Em geral, independente da quantidade de símbolos nas expressões, o melhor desempenho para a associação correta média de símbolos é obtido com $\gamma = 0.25$, apresentando percentuais de acerto entre 90% e 100%. A relevância do *shape context* local pode ser observada quando se compara $\gamma > 0$ (combinação de *shape context* global e local) com $\gamma = 0$ (apenas *shape context* global).

Custos combinados

Ao compararmos a melhor associação correta média global obtida considerando apenas o custo de aresta ($\beta = 1.0$ e grafo completo) com a obtida considerando somente o custo de vértice ($\gamma = 0.25$), observa-se que os resultados são próximos, isto é, 98,3% contra 98,1% respectivamente. Conforme já informado, esses valores dos parâmetros β e γ foram fixados para avaliação do método com

Figura 5.10: Associação correta média global por casamento obtida com $\gamma = \{0.0, 0.25, 0.5, 0.75\}$ para o Conjunto B.

diferentes valores para o parâmetro α . Nessa avaliação foi considerada apenas a conectividade de grafo completo para as arestas.

A Figura 5.11 apresenta a associação correta média global por casamento obtida com variações de α em 0.10. Os melhores resultados são obtidos com $\alpha = \{0.2, 0.3, 0.4\}$, os quais são detalhados na Figura 5.12. No geral, a melhor associação correta média é obtida com $\alpha = 0.3$, apesar de que, em determinadas expressões, obtém-se um desempenho superior com outros valores para α . É interessante notar que, quando utiliza-se $\alpha = 0.3$, o desempenho por expressão é sempre superior a 97,5%.

Figura 5.11: Associação correta média global por casamento obtida com $\alpha = [0.0, 1.0]$ para o Conjunto B, com $\beta = 1.0$ e $\gamma = 0.25$.

Figura 5.12: Associação correta média por casamento obtida com $\alpha = \{0.2, 0.3, 0.4\}$ para o Conjunto B, com $\beta = 1.0$ e $\gamma = 0.25$.

A associação correta média global por casamento obtida com a combinação dos custos apresenta melhor eficácia do que os casamentos que consideram o custo de aresta e de vértice isoladamente, ou seja, 99,7% contra 98,3% e 98,1% respectivamente. Essa comparação é apresentada na Figura 5.13. Enquanto o desempenho obtido com os custos de vértice e aresta isolados é relativamente menor para expressões com maior quantidade de símbolos, o desempenho obtido com a combinação dos custos sofre pouca alteração.

As Figuras 5.14, 5.15 e 5.16, apresentam o desvio padrão obtido com cada custo para comparação.

(a) Resultados com variações do custo combinado, (b) Resultados normalizados com variações do custo combinado, vértice e aresta.

Figura 5.13: Comparação entre a associação correta média por casamento obtida com custo combinado, vértice e aresta.

Os resultados obtidos considerando-se apenas o custo de aresta apresentam variâncias muito altas para expressões com mais do que 24 símbolos. Quando apenas o custo de vértice é considerado, no geral, as variâncias apresentam um aspecto uniforme, independente da quantidade de símbolos presente em cada classe de expressão.

Quando o custo combinado é considerado, os resultados apresentam, de um modo geral, variâncias inferiores aos resultados obtidos com a utilização do custo de aresta e vértice isolados. As expressões com maiores quantidades de símbolos apresentam as maiores variâncias.

(a) Desvio padrão obtido com o custo de aresta ótimo.

(b) Desvio padrão normalizado obtido com o custo de aresta ótimo.

Figura 5.14: Desvio padrão da associação correta média obtido com o custo de aresta ($\beta = 1.0$).

(a) Desvio padrão obtido com o custo de vértice ótimo.

(b) Desvio padrão normalizado obtido com o custo de vértice ótimo.

Figura 5.15: Desvio padrão da associação correta média obtido com o custo de vértice ($\gamma = 0.25$).

(a) Desvio padrão obtido com o custo combinado ótimo.

(b) Desvio padrão normalizado obtido com o custo combinado ótimo.

Figura 5.16: Desvio padrão da associação correta média obtido com o custo combinado ($\alpha = 0.3$, $\beta = 1.0$ e $\gamma = 0.25$).

As Tabelas 5.2 e 5.3 apresentam a associação correta média por casamento e o respectivo desvio padrão. Analisando os resultados, observa-se que a técnica apresentou resultados superiores a 98,7% para todas as classes de expressões, com exceção das classes 27, 51 e 56, com porcentagem média de acertos de 97,91%, 98,14% e 97,61% respectivamente. O maior desvio padrão é obtido na classe 27 (8,39%).

Tabela 5.2: Resultados dos casamentos para as classes de expressão 1 ~ 28, obtidos com o custo combinado ótimo ($\alpha = 0.3$, $\beta = 1.0$ e $\gamma = 0.25$)

Classe	Qtd. Instâncias	Qtd. Símbolos	Média	Desv. Padrão	Média (%)	Desv. Padrão (%)
1	17	4	4.00	0.00	100.00%	0.00%
2	18	7	7.00	0.00	100.00%	0.00%
3	18	7	7.00	0.00	100.00%	0.00%
4	18	8	8.00	0.00	100.00%	0.00%
5	19	9	9.00	0.00	100.00%	0.00%
6	18	9	9.00	0.00	100.00%	0.00%
7	19	10	10.00	0.00	100.00%	0.00%
8	17	10	10.00	0.00	100.00%	0.00%
9	17	12	12.00	0.00	100.00%	0.00%
10	16	13	13.00	0.00	100.00%	0.00%
11	17	13	12.99	0.12	99.94%	0.93%
12	17	13	13.00	0.00	100.00%	0.00%
13	15	13	13.00	0.00	100.00%	0.00%
14	17	13	13.00	0.00	100.00%	0.00%
15	19	14	14.00	0.00	100.00%	0.00%
16	18	14	14.00	0.00	100.00%	0.00%
17	16	15	14.98	0.22	99.83%	1.48%
18	16	16	15.99	0.13	99.95%	0.81%
19	22	16	16.00	0.00	100.00%	0.00%
20	19	16	15.99	0.15	99.93%	0.95%
21	15	16	16.00	0.00	100.00%	0.00%
22	18	18	18.00	0.00	100.00%	0.00%
23	18	18	18.00	0.00	100.00%	0.00%
24	18	19	18.88	0.50	99.38%	2.62%
25	17	19	19.00	0.00	100.00%	0.00%
26	18	20	20.00	0.00	100.00%	0.00%
27	20	20	19.58	1.68	97.91%	8.39%
28	19	20	19.98	0.22	99.88%	1.08%

Tabela 5.3: Resultados dos casamentos para as classes de expressão 29 ~ 56, obtidos com o custo combinado ótimo ($\alpha = 0.3$, $\beta = 1.0$ e $\gamma = 0.25$)

Classe	Qtd. Instâncias	Qtd. Símbolos	Média	Desv. Padrão	Média (%)	Desv. Padrão (%)
29	18	20	19.93	0.37	99.64%	1.86%
30	17	20	20.00	0.00	100.00%	0.00%
31	16	21	21.00	0.00	100.00%	0.00%
32	16	22	21.75	0.66	98.86%	3.01%
33	17	22	22.00	0.00	100.00%	0.00%
34	17	22	22.00	0.00	100.00%	0.00%
35	17	23	23.00	0.00	100.00%	0.00%
36	17	23	23.00	0.00	100.00%	0.00%
37	16	24	23.81	0.59	99.20%	2.45%
38	17	24	23.99	0.17	99.94%	0.71%
39	15	25	24.80	1.14	99.22%	4.58%
40	15	25	24.98	0.19	99.92%	0.78%
41	15	25	24.96	0.27	99.85%	1.09%
42	16	27	26.98	0.18	99.94%	0.67%
43	17	28	27.73	0.72	99.03%	2.58%
44	18	28	28.00	0.00	100.00%	0.00%
45	16	31	30.99	0.13	99.97%	0.42%
46	19	31	30.62	1.21	98.76%	3.90%
47	17	31	30.68	0.73	98.98%	2.35%
48	19	32	32.00	0.00	100.00%	0.00%
49	18	32	31.83	0.58	99.47%	1.81%
50	16	39	38.60	0.93	98.97%	2.38%
51	15	41	40.24	1.45	98.14%	3.53%
52	17	45	44.71	2.12	99.35%	4.70%
53	14	49	48.81	0.72	99.62%	1.46%
54	16	49	48.37	1.28	98.71%	2.61%
55	18	53	52.65	0.84	99.35%	1.58%
56	12	66	64.42	3.67	97.61%	5.57%

Características de boas expressões como modelos

Dois aspectos interessantes para se avaliar são as qualidades e características para uma expressão ser utilizada como modelo. Pode-se dizer que, para ser considerada como um bom modelo, a expressão deve ter, em média, uma grande quantidade de acertos, isto é, associações corretas de símbolos.

Podemos avaliar características de bons modelos a partir dos resultados obtidos com o método proposto sobre o conjunto B, uma vez que um esquema de casamento cruzado foi empregado e, assim, todas as expressões do conjunto foram utilizadas como modelo em algum momento.

Um levantamento a partir dos resultados obtidos com $\alpha = 0.3$ em todas as classes de expressões é apresentado na Tabela 5.4. Para cada classe, são apresentadas a quantidade de expressões categorizadas por intervalos de porcentagem de acerto do método quando utilizadas como modelo. Foram escolhidos cinco intervalos de porcentagem de acerto a saber: $[85\%,90\%]$, $[90\%,95\%]$, $[95\%, 98\%]$ e $[98\%,100\%]$. A partir do apresentado, observa-se que, das 957 expressões, 914 estão localizadas no intervalo $[98\%,100\%]$ e 31 expressões no intervalo $[95\%, 98\%]$. Deste modo, o método apresenta resultados entre 95% e 100% para 945 (98,74%) das expressões.

A avaliação sobre classes com grande quantidade expressões que apresentam somente altas taxas de acerto não é interessante, pois é difícil avaliar aspectos que caracterizem bons ou maus modelos. Assim, a análise foi concentrada em classes que apresentaram tanto bons casamentos quanto ruins. Em particular, foram consideradas 6 classes com 11 expressões que apresentaram desempenhos de até 95% de acerto. Primeiramente, analisaremos expressões da classe 27 e, depois, expressões das classes 39, 46, 51, 52, 56. A classe 32, que apresenta uma expressão no intervalo $[90\%,95\%]$, não foi considerada na análise.

Tabela 5.4: Resultados do casamento categorizados por intervalos de desempenho obtido com o custo combinado ótimo ($\alpha = 0.3$)

Classe	# Símbolos.	# Instancias	[85%,90%]	[90%,95%]	[95%, 98%]	[98%,100%]
1	4	17				17
2	7	18				18
3	7	18				18
4	8	18				18
5	9	19				19
6	9	18				18
7	10	19				19
8	10	17				17
9	12	17				17
10	13	16				16
11	13	17				17
12	13	17				17
13	13	15				15
14	13	17				17
15	14	19				19
16	14	18				18
17	15	16				16
18	16	16				16
19	16	22				22
20	16	19				19
21	16	15				15
22	18	18				18
23	18	18				18
24	19	18			1	17
25	19	17				17
26	20	18				18
27	20	20		6		14
28	20	19				19
29	20	18			1	17
30	20	17				17
31	21	16				16
32	22	16		1		15
33	22	17				17
34	22	17				17
35	23	17				17
36	23	17				17
37	24	16			3	13
38	24	17				17
39	25	15		1	1	13
40	25	15				15
41	25	15				15
42	27	16				16
43	28	17			2	15
44	28	18				18
45	31	16				16
46	31	19		1	2	16
47	31	17			4	13
48	32	19				19
49	32	18				18
50	39	16			2	14
51	41	15		1	6	8
52	45	17		1	1	15
53	49	14				14
54	49	16			4	12
55	53	18			1	17
56	66	12	1		3	8
TOTAL		957	1	11	31	914

A Figura 5.17 apresenta uma expressão da classe 27 com bom desempenho no casamento cruzado (média de 100% associações corretas e desvio padrão de 0%) e a Figura 5.18 apresenta expressões da mesma classe onde o método de casamento obteve os desempenhos mais baixos. Comparando-se as expressões da Figura 5.18 com a da Figura 5.17, nota-se certas irregularidades estruturais nas expressões com menor desempenho que dificultam o casamento, tais como: espaçamentos não-uniformes entre os símbolos e numeradores e denominadores desalinhados em relação a fração.

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

Figura 5.17: Expressão da classe 27 onde o método de casamento obteve maior desempenho.

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(a) Espaçamento entre símbolos não uniforme no denominador.

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(b) Numerador desalinhado em relação ao traço da fração

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(c) Numerador e denominador desalinhados com o traço da fração

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(d) Espaçamento curto entre os símbolos na parte direita do denominador

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(e) Numerador desalinhado em relação ao traço da fração

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(f) Espaçamento curto entre os símbolos na parte direita do denominador

Figura 5.18: Expressões da classe 27 onde o método de casamento obteve menor desempenho como modelo.

Para ilustrar as dificuldades mencionadas, a Figura 5.19 apresenta somente as associações incorretas obtidas nas expressões da classe 27 (a expressão modelo está acima da expressão de entrada). A expressão de entrada da Figura 5.19(a) é apresentada como expressão modelo na Figura 5.19(b) e vice-versa.

Claramente, observa-se numeradores desalinhados em relação a fração, o que pode causar asso-

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(a) Casamento com 11 (55%) símbolos associados corretamente.

$$A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(b) Casamento com 9 (45%) símbolos associados corretamente.

Figura 5.19: Casamento de expressões da classe 27.

ciações incorretas de símbolos localizados nos numeradores para símbolos situados no denominador. O mal posicionamento dos símbolos pode ter gerado associações entre símbolos diferentes. Outro exemplo de casamento da classe 27 é apresentado na Figura 5.20, na qual observa-se espaçamento não-uniforme dos símbolos no denominador

As Figuras 5.21, 5.22 ilustram respectivamente as expressões das classes 39 e 46 nas quais o método de casamento apresentou o pior e melhor desempenho. Na expressão da classe 39 com pior desempenho, observa-se que o espaçamento dos símbolos no denominador é maior do que no numerador, diferente do visto na expressão com melhor desempenho, com espaçamento uniforme entre os símbolos e alinhados em relação à fração.

$$A_{2K} = \frac{2RA_K}{2R + \sqrt{4R^2 + A_K^2}}$$

$$A_{z_k} = \frac{2RA_K}{2R + \sqrt{4R^2 + A_K^2}}$$

Figura 5.20: Casamento de uma expressão da classe 27 com 13 (65%) símbolos associados corretamente.

$$y = \frac{x' \sin \theta + y' \sin(w' + \theta)}{\sin w} \quad y = \frac{x' \sin \theta + y' \sin(w' + \theta)}{\sin w}$$

(a) Pior desempenho: símbolos espaçados no denominador.

(b) Melhor desempenho.

Figura 5.21: Expressões da classe 39.

$$\frac{b^2c^2 - 4b^3d - 4ac^3 + 18abcd - 27a^2d^2}{a^4} \quad \frac{b^2c^2 - 4b^3d - 4ac^3 + 18abcd - 27a^2d^2}{a^4}$$

(a) Pior desempenho: espaçamento não uniforme. À medida que a escrita avança para a direita, o espaçamento diminui.

(b) Melhor desempenho.

Figura 5.22: Expressões da classe 46.

Na expressão da classe 46 com pior desempenho nota-se que o espaçamento entre os símbolos é relativamente menor na parte direita do numerador da expressão modelo. A expressão da mesma classe com melhor desempenho apresenta os símbolos alinhados em relação à fração.

Um casamento de expressões da classe 39 é apresentada na Figura 5.23 e da classe 46 é apresentada na Figura 5.24 (somente as associações incorretas para ambos os casamentos). No primeiro casamento, o numerador na expressão modelo não está centralizado, o que pode ter influenciado na associação entre símbolos. Já no segundo, os símbolos estão “exprimidos” na parte direita do numerador da expressão modelo, dificultando o casamento.

The figure shows two handwritten versions of the equation $y = \frac{x' \sin \theta + y' \sin(\omega' + \theta)}{\sin \omega}$. The top version is the 'model' expression, and the bottom version is the 'candidate' expression. Lines connect corresponding symbols between the two: x' to x' , $\sin \theta$ to $\sin \theta$, $+$ to $+$, y' to y' , $\sin(\omega' + \theta)$ to $\sin(\omega' + \theta)$, and the denominator $\sin \omega$ to $\sin \omega$. The mapping is correct for all symbols.

Figura 5.23: Casamento de expressões da classe 39 com 12 (52,17%) símbolos mapeados corretamente.

The figure shows two handwritten versions of the equation $b^2 c^2 - 4b^3 d - 4ac^3 + 18abcd - 27a^2 d^2$. The top version is the 'model' expression, and the bottom version is the 'candidate' expression. Lines connect corresponding symbols between the two: b^2 to b^2 , c^2 to c^2 , $-$ to $-$, $4b^3 d$ to $4b^3 d$, $-$ to $-$, $4ac^3$ to $4ac^3$, $+$ to $+$, $18abcd$ to $18abcd$, $-$ to $-$, and $27a^2 d^2$ to $27a^2 d^2$. The mapping is correct for all symbols.

Figura 5.24: Casamento de expressões da classe 46 com 27 (87,09%) símbolos mapeados corretamente.

As Figuras 5.25, 5.26 e 5.27 apresentam respectivamente as expressões das classes 51, 52 e 56 nas quais o método de casamento apresentou o pior e o melhor desempenho. Novamente, nota-se que as expressões com piores desempenhos envolvem espaçamentos não uniformes, numeradores e denominadores não alinhados em relação ao traço da fração. Já as expressões com melhores desempenhos mostram-se com espaçamentos mais uniformes e alinhados em relação à fração.

Pelos resultados apresentados, pode-se dizer que bons modelos devem ter as seguintes características:

- espaçamento uniforme entre os símbolos na expressão como um todo;
- numeradores alinhados com denominadores e ambos centralizados em relação ao traço da fração;
- traços dos símbolos bem definidos, isto é, traços claros e sem falhas.

$$r(r_0 \sin(\theta - \theta_0) - r_1 \sin(\theta - \theta_1)) = r_0 r_1 \sin(\theta_1 - \theta_0) \quad r(r_0 \sin(\theta - \theta_0) - r_1 \sin(\theta - \theta_1)) = r_0 r_1 \sin(\theta_1 - \theta_0)$$

(a) Pior desempenho: pouco espaçamento dos símbolos na parte direita da expressão. (b) Melhor desempenho.

Figura 5.25: Expressões da classe 51.

$$\cos^{-1} = \frac{a_0 a_1 + b_0 b_1 + c_0 c_1}{\sqrt{a_0^2 + b_0^2 + c_0^2} \sqrt{a_1^2 + b_1^2 + c_1^2}} \quad \cos^{-1} = \frac{a_0 a_1 + b_0 b_1 + c_0 c_1}{\sqrt{a_0^2 + b_0^2 + c_0^2} \sqrt{a_1^2 + b_1^2 + c_1^2}}$$

(a) Pior desempenho: símbolos no denominador desalinhados com a fração. (b) Melhor desempenho

Figura 5.26: Expressões da classe 52.

$$\begin{array}{c}
\begin{array}{c}
\begin{array}{ccc}
x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\
a_0 & b_0 & c_0 \\
a_1 & b_1 & c_1
\end{array} \\
\hline
\sqrt{\begin{array}{c} |b_0 \ c_0|^2 + |c_0 \ a_0|^2 + |a_0 \ b_0|^2 \\ |b_1 \ c_1|^2 + |c_1 \ a_1|^2 + |a_1 \ b_1|^2 \end{array}}
\end{array}
\end{array}
\quad
\begin{array}{c}
\begin{array}{ccc}
x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\
a_0 & b_0 & c_0 \\
a_1 & b_1 & c_1
\end{array} \\
\hline
\sqrt{\begin{array}{c} |b_0 \ c_0|^2 + |c_0 \ a_0|^2 + |a_0 \ b_0|^2 \\ |b_1 \ c_1|^2 + |c_1 \ a_1|^2 + |a_1 \ b_1|^2 \end{array}}
\end{array}
\end{array}$$

(a) Pior desempenho: símbolos no denominador desalinhados com a fração.

(b) Melhor desempenho

Figura 5.27: Expressões da classe 56.

5.3 Validação

Uma vez que os valores dos parâmetros foram escolhidos baseados em uma avaliação sobre o conjunto B, poderia haver, em princípio, um forte viés nessa escolha. Para evidenciar que a escolha é, de fato, apropriada, os parâmetros ótimos $\beta = 1.0$, $\gamma = 0.25$ e $\alpha = 0.3$ obtidos anteriormente foram utilizados para validação do método sobre o conjunto de dados A.

A associação correta média global por casamento apresentou um índice de acerto de 99,8%. Para conferir se, de fato, o valor $\alpha = 0.3$ é uma boa escolha, outros valores para o parâmetro α foram testados no método e os resultados são apresentados na Figura 5.28. Nota-se que a melhor associação correta média global (99,9%) é obtida com $\alpha = 0.4$, bem próximo de $\alpha = 0.3$. Esses resultados demonstram que o método (juntamente com sua parametrização) é robusto para outro conjunto de expressões matemáticas.

Figura 5.28: Associação correta média global por casamento obtida com $\alpha = [0.0, 1.0]$ para o Conjunto A.

Capítulo 6

Conclusões

Neste trabalho, foi proposta uma abordagem para tratar o problema de rotular os símbolos de uma expressão matemática manuscrita. Mais especificamente, a abordagem consiste em um método que, dada uma expressão modelo (cujos símbolos já encontram-se previamente segmentados e rotulados) e uma transcrição da mesma (com símbolos também segmentados, porém não rotulados), realiza a associação dos símbolos da transcrição aos correspondentes símbolos da expressão modelo. Esse problema de associação foi denominado de casamento de expressões.

Dentre as contribuições deste trabalho, estão a formulação do casamento de expressões como um problema de associação simples e, também, a especificação de um custo de associação que considera tanto aspectos estruturais globais da expressão quanto aspectos locais associados aos símbolos matemáticos. Além disso, foi adotado o método de segmentação durante a escrita, no qual os usuários são induzidos a produzirem uma escrita na qual os símbolos já são corretamente segmentados.

Foram realizados experimentos em um conjunto composto de 957 expressões e obtida uma taxa média de acerto de 99,7%. Uma validação da técnica foi realizada com um outro conjunto de 64 expressões, obtendo-se uma taxa média de acerto de 99,9%, evidenciando-se a robustez da técnica.

Os esforços realizados neste trabalho resultaram em um artigo, um software e um grande conjunto de dados rotulado, que pode ser facilmente ampliado. A partir do apresentado, constata-se que o método proposto pode ser utilizado para o propósito considerado neste trabalho. As implementações da abordagem deste trabalho estão disponíveis em <http://code.google.com/p/express-match/>.

6.1 Trabalhos Futuros

A partir dos resultados, constata-se que as maiores dificuldades no casamento de expressões estão em expressões com maiores quantidades de símbolos. A seguir são descritas algumas oportunidades e desafios para aprimorar a técnica proposta.

Segmentação de símbolos: uma restrição forte da abordagem proposta é a suposição de que os símbolos estejam corretamente segmentados antes do casamento. Tal suposição limita a sua

aplicabilidade a dispositivos tipo *tablet* e não permite o uso em outros dispositivos *pen-based* e *touch-screen*. O desafio, neste caso, é considerar somente os traços da escrita do usuário, permitindo que o mesmo possa inserir ou remover traços independente da ordem de escrita. Para viabilizar isso, seria necessário considerar um problema mais difícil, como, por exemplo, o mapeamento de traço para traço, traço para símbolo, combinações de traços, etc.

Esquema de votação para o algoritmo de casamento: expressões de entrada rotuladas pelo algoritmo podem ser utilizadas como novas expressões modelo. O algoritmo de casamento pode ser modificado para considerar mais de uma expressão modelo e um esquema de votação pode ser definido para a rotulação dos símbolos. Consequentemente, o algoritmo considerará mais variações de expressões, tornando-se mais robusto.

Combinação com técnicas para análise estrutural: a associação entre símbolos pode ser realizada entre casamento de estruturas de expressões matemáticas atrelada a um custo como, por exemplo, o de deformação estrutural. Tais estruturas podem ser estabelecidas por meio de técnicas para análise estrutural de expressões matemáticas, tais como *Baseline Structure Tree* [53] e outras técnicas mencionadas na seção 3.2.3.

Aprimorar o espaçamento e alinhamento entre os símbolos das expressões: foram identificadas irregularidades estruturais que dificultam o casamento no capítulo 5, tais como espaçamentos não-uniformes entre os símbolos e numeradores e denominadores desalinhados em relação a fração. Algoritmos e heurísticas podem ser aplicados para amenizar ou eliminar as irregularidades estruturais existentes nas expressões matemáticas manuscritas, como, por exemplo, delimitar um determinado espaço entre os símbolos. Isso certamente reduzirá problemas de casamento de expressões, facilitando o casamento correto dos símbolos.

Utilização de outras medidas de similaridade: os custos de casamento podem considerar outras medidas de similaridade, tais como o Casamento Elástico [51], *Dynamic Time Warping* e *Edit Distance*. Assim como o *shape context*, essas medidas consideram mudanças estruturais dos símbolos e podem auxiliar na associação correta.

Apêndice A

Expressões do conjunto de dados B

A seguir são apresentados exemplos de cada classe de expressões matemáticas do conjunto de dados B, mencionado na seção 5.1.2.

$$n! - 1 \quad x = r \cos \theta \quad y = r \sin \theta \quad y^4 + y + 1 = 0$$

(a) Classe 01

(b) Classe 02

(c) Classe 03

(d) Classe 04

$$a^x a^y = a^{x+y} \quad a \equiv b \pmod{n} \quad \Theta = \tan^{-1} \frac{y}{x}$$

(e) Classe 05

(f) Classe 06

(g) Classe 07

$$A + B + C = 180^\circ \quad v_1 = \sqrt{D} + \sqrt{D - \epsilon \sqrt{D}}$$

(h) Classe 08

(i) Classe 09

$$\sin \theta = \frac{y}{\sqrt{x^2 + y^2}} \quad \cos \theta = \frac{x}{\sqrt{x^2 + y^2}}$$

(j) Classe 10

(k) Classe 11

$$a = b \cos C + c \cos B \quad b = c \sin B = c \cos A$$

(l) Classe 12

(m) Classe 13

Figura A.1: Exemplos de expressões do conjunto de dados B (Classes 01 a 13).

$$\sqrt[x]{\frac{a}{b}} = \frac{\sqrt[x]{a}}{\sqrt[x]{b}}$$

(a) Classe 14

$$x' = x \cos \Theta + y \sin \Theta$$

(b) Classe 15

$$P(x) = \sum_{i=0}^n a_j x^j$$

(c) Classe 16

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(d) Classe 17

$$c^2 = a^2 + b^2 - 2ab \cos C$$

(e) Classe 18

$$p = \sqrt{a^2 + b^2 - 2ab \cos A}$$

(f) Classe 19

$$\cos \alpha = \frac{a}{\sqrt{a^2 + b^2 + c^2}}$$

(g) Classe 20

$$1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

(h) Classe 21

$$\left| \frac{ax_0 + by_0 + c}{\sqrt{a^2 + b^2}} \right|$$

(i) Classe 22

$$a^2 + b^2 = (a + bi)(a - bi) \quad q = \sqrt{\left(\frac{B-A}{C}\right)^2 + 1} + \frac{B-A}{C}$$

(j) Classe 23

(k) Classe 24

$$\left(\frac{a}{n}\right) = \prod_{i=1}^k \left(\frac{a}{p_i}\right)^{b_i}$$

(l) Classe 25

Figura A.2: Exemplos de expressões do conjunto de dados B (Classes 14 a 25).

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} \quad A_{2k} = \frac{2RA_k}{2R + \sqrt{4R^2 + A_k^2}}$$

(a) Classe 26

(b) Classe 27

$$S = \left(\sum_{i=1}^n \theta_i - (n-2)\pi \right) r^2 \quad |P|_2 = \left(\sum_{j=1}^n |a_j|^2 \right)^{1/2} \quad \sum_{i=1}^k \frac{x^{a_i}}{1-x^{b_i}} = \frac{1}{1-x}$$

(c) Classe 28

(d) Classe 29

(e) Classe 30

$$\left(a \frac{1-t^2}{1+t^2}, \frac{2bt}{1+t^2} \right) \quad \frac{1}{r^2} = \frac{1}{(R-m)^2} + \frac{1}{(R+m)^2}$$

(f) Classe 31

(g) Classe 32

$$\frac{1}{\frac{1}{2} \left(\frac{1}{a} + \frac{1}{b} \right)} = \frac{2ab}{a+b} \quad \sqrt[n]{1} = \cos \frac{2K\pi}{n} + i \sin \frac{2K\pi}{n}$$

(h) Classe 33

(i) Classe 34

$$\frac{a}{b} < \frac{c}{d} \Rightarrow \frac{a}{b} < \frac{a+c}{b+d} < \frac{c}{d} \quad \frac{1}{4} + \left(\frac{1}{4} \right)^2 + \left(\frac{1}{4} \right)^3 + \dots = \frac{1}{3}$$

(j) Classe 35

(k) Classe 36

$$\Phi_p(x) = \sum_{k=0}^{p-1} x^k = \frac{x^p - 1}{x - 1}$$

(l) Classe 37

Figura A.3: Exemplos de expressões do conjunto de dados B (Classes 26 a 37).

$$\left| x - \frac{p_n}{q_n} \right| \leq \frac{1}{q_n q_{n+1}} < \frac{1}{q_n^2} \quad y = \frac{x' \sin \theta + y' \sin(\omega' + \theta)}{\sin \omega}$$

(a) Classe 38

(b) Classe 39

$$y = aS(t) = a \int_0^t \sin\left(\frac{1}{2} \pi s^2\right) ds$$

(c) Classe 40

$$y^2 \pm \sqrt{u-p} \left(y - \frac{q}{2(u-p)} \right) + \frac{u}{2} = 0$$

(d) Classe 41

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

(e) Classe 42

$$V_n(r) \approx \frac{n^{-(n+1)/2}}{\sqrt{\pi}} (2\pi e)^{n/2} r^n \quad \|P\|_2 = \left(\int_0^{2\pi} |P(e^{i\theta})|^2 \frac{d\theta}{2\pi} \right)^{1/2}$$

(f) Classe 43

(g) Classe 44

$$\Delta = \begin{vmatrix} A & \frac{1}{2}C & \frac{1}{2}D \\ \frac{1}{2}C & B & \frac{1}{2}E \\ \frac{1}{2}D & \frac{1}{2}E & F \end{vmatrix} \quad \frac{b^2 c^2 - 4b^3 d - 4ac^3 + 18abcd - 27a^2 d^2}{a^4}$$

(h) Classe 45

(i) Classe 46

$$a_1 + a_2 + \dots + a_n = \begin{cases} a_1 \frac{1-r^n}{1-r} & r \neq 1 \\ na_1 & r = 1 \end{cases} \quad \frac{r(x)}{q(x)} = \frac{Ax + B}{x^2 + bx + c} + \frac{s(x)}{h(x)}$$

(j) Classe 47

(k) Classe 48

Figura A.4: Exemplos de expressões do conjunto de dados B (Classes 38 a 48).

$$\sigma_k(n) = \sum_{d|n} d_k = \prod_i \frac{p_i^{k(\alpha_i+1)} - 1}{p_i^k - 1}$$

(a) Classe 49

$$a \int_0^\theta \sqrt{1 - e^2 \cos^2 \phi} d\phi = a \left(E\left(\frac{\pi}{2}, e\right) - E\left(\frac{\pi}{2} - \theta, e\right) \right)$$

(b) Classe 50

$$r(r_0 \sin(\theta - \theta_0) - r_1 \sin(\theta - \theta_1)) = r_0 r_1 \sin(\theta_1 - \theta_0)$$

(c) Classe 51

$$\cos^{-1} = \frac{a_0 a_1 + b_0 b_1 + c_0 c_1}{\sqrt{a_0^2 + b_0^2 + c_0^2} \sqrt{a_1^2 + b_1^2 + c_1^2}} \quad \begin{bmatrix} -r^2 y_0 & r^2 x_0 & 0 & 0 \\ -r x_0 z_0 & -r y_0 z_0 & r p^2 & 0 \\ 0 & 0 & 0 & r p \\ -p x_0 & -p y_0 & -p z_0 & r^2 p \end{bmatrix}$$

(d) Classe 52

(e) Classe 53

$$M(p) = |a_n| \prod_{i=1}^n \max(1, |z_i|) = \exp\left(\int_0^1 \log |P(e^{2\pi i t})| dt\right)$$

(f) Classe 54

$$M = \begin{bmatrix} -a^2 + b^2 + c^2 & -2ab & -2ac \\ -2ab & a^2 - b^2 + c^2 & -2bc \\ -2ac & -2bc & a^2 + b^2 - c^2 \end{bmatrix}$$

(g) Classe 55

$$\frac{\begin{vmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \end{vmatrix}}{\sqrt{\begin{vmatrix} b_0 & c_0 \\ b_1 & c_1 \end{vmatrix}^2 + \begin{vmatrix} c_0 & a_0 \\ c_1 & a_1 \end{vmatrix}^2 + \begin{vmatrix} a_0 & b_0 \\ a_1 & b_1 \end{vmatrix}^2}}$$

(h) Classe 56

Figura A.5: Exemplos de expressões do conjunto de dados B (Classes 49 a 56).

Referências Bibliográficas

- [1] F. D. J. Aguilar and N. S. T. Hirata, *ExpressMatch: a system for creating ground-truthed datasets of online mathematical expressions*, 10th IAPR International Workshop on Document Analysis Systems, 2012, pp. 1–5. [ix](#), [x](#), [3](#), [28](#), [43](#)
- [2] Ahmad-Montaser Awal, Harold Mouchere, and Christian Viard-Gaudin, *Towards handwritten mathematical expression recognition*, Proceedings of the 2009 10th International Conference on Document Analysis and Recognition (Washington, DC, USA), ICDAR '09, IEEE Computer Society, 2009, pp. 1046–1050. [1](#), [20](#), [21](#), [25](#), [26](#), [27](#)
- [3] ———, *The problem of handwritten mathematical expression recognition evaluation*, Proceedings of the 2010 12th International Conference on Frontiers in Handwriting Recognition (Washington, DC, USA), ICFHR '10, IEEE Computer Society, 2010, pp. 646–651. [1](#), [2](#), [23](#), [25](#), [26](#)
- [4] S. Belongie, J. Malik, and J. Puzicha, *Shape matching and object recognition using shape contexts*, IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002), 509–522. [ix](#), [9](#), [15](#), [16](#), [17](#)
- [5] E. Bengoetxea, *Inexact graph matching using estimation of distribution algorithms*, Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, December 2002. [5](#)
- [6] J.A. Bondy and U.S.R. Murty, *Graph theory*, Springer, 2008. [5](#)
- [7] Kam-Fai Chan and Dit-Yan Yeung, *Mathematical expression recognition: a survey*, International Journal on Document Analysis and Recognition **3** (2000), 3–15, 10.1007/PL00013549. [ix](#), [20](#), [22](#), [23](#), [25](#)
- [8] Kam-Fai Chan and Dit-Yan Yeung, *Error detection, error correction and performance evaluation in on-line mathematical expression recognition*, Pattern Recognition **34** (2001), no. 8, 1671 – 1684. [26](#)
- [9] Rommel Teodoro de Oliveira, *Sobre conjuntos dominantes eficientes em grafos*, Master's thesis, Universidade Federal de Goiás, 2009. [ix](#), [7](#)
- [10] Luiz Henrique de Figueiredo and Paulo Cezar Pinto Carvalho, *Notas de geometria computacional*, 2009, <http://lhf.impa.br/cursos/gc/notas.pdf>, [Online; acesso 30 de dezembro de 2011]. [12](#)
- [11] John A. Fitzgerald, Franz Geiselbrechtinger, and Tahar Kechadi, *Structural analysis of handwritten mathematical expressions through fuzzy parsing*, Proceedings of the 2nd IASTED international conference on Advances in computer science and technology (Anaheim, CA, USA), ACTA Press, 2006, pp. 151–156. [23](#)

- [12] Naohiro Furukawa, Junko Tokuno, and Hisashi Ikeda, *Online character segmentation method for unconstrained handwriting strings using off-stroke features*, Tenth International Workshop on Frontiers in Handwriting Recognition (La Baule (France)) (Guy Lorette, ed.), Université de Rennes 1, Suvisoft, October 2006, <http://www.suvisoft.com> Université de Rennes 1 (English). 21, 22
- [13] Utpal Garain and B. B. Chaudhuri, *Recognition of online handwritten mathematical expressions*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (2004), 2366–2376. 22
- [14] Ray Genoe and Tahar Kechadi, *Fuzzy spatial analysis techniques for mathematical expression recognition*, Proceedings of the 10th international conference on Artificial intelligence and soft computing: Part I (Berlin, Heidelberg), ICAISC'10, Springer-Verlag, 2010, pp. 80–87. 20, 26
- [15] Andre Luiz Pires Guedes, *Triangulação*, 1996, <http://www.inf.ufpr.br/andre/geom/node8.html>, [Online; acesso 2 de fevereiro de 2012]. ix, 12, 13
- [16] G. Guida and G. Mauri, *Evaluation of natural language processing systems: Issues and approaches*, Proceedings of the IEEE 74 (1986), no. 7, 1026 – 1035. 25
- [17] Isabelle Guyon, Lambert Schomaker, Rejean Plamondon, Mark Liberman, and Stan Janet, *Unipen project of on-line data exchange and recognizer benchmarks*, International Conference on Pattern Recognition, 1994. 1
- [18] Nina S. T. Hirata and Willian Y. Honda, *Automatic labeling of handwritten mathematical symbols via expression matching*, Proceedings of the 8th international conference on Graph-based representations in pattern recognition (Berlin, Heidelberg), GbRPR'11, Springer-Verlag, 2011, pp. 295–304. 3, 21, 35, 41
- [19] B. Q. Huang, Y. B. Zhang, and M. T. Kechadi, *Preprocessing techniques for online handwriting recognition*, Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications (Washington, DC, USA), ISDA '07, IEEE Computer Society, 2007, pp. 793–800. 37
- [20] B.Q Huang and M-T Kechadi, *A structural analysis approach for online handwritten mathematical expressions*, International Journal of Computer Science and Network Security, IJCSNS '07, 2007. 1, 20, 23
- [21] C. V. Jawahar, A. Balasubramanian, Million Meshesha, and Anoop M. Namboodiri, *Retrieval of online handwriting by synthesis and matching*, Pattern Recogn. 42 (2009), no. 7, 1445–1457. ix, 23
- [22] Dae Hwan Kim and Jin H. Kim, *Top-down search with bottom-up evidence for recognizing handwritten mathematical expressions*, Proceedings of the 2010 12th International Conference on Frontiers in Handwriting Recognition (Washington, DC, USA), ICFHR '10, IEEE Computer Society, 2010, pp. 507–512. 27
- [23] Kang Kim, Taik Heon Rhee, Jae Seung Lee, and Jin Hyung Kim, *Utilizing consistency context for handwritten mathematical expression recognition*, Proceedings of the 2009 10th International Conference on Document Analysis and Recognition (Washington, DC, USA), ICDAR '09, IEEE Computer Society, 2009, pp. 1051–1055. 2, 26
- [24] Harold W Kuhn, *The hungarian method for the assignment problem*, Naval Res. Logist. Quart. 2 (1955), 83–97. 9

- [25] Adrien Lapointe, *Issues in performance evaluation of mathematical notation recognition systems*, Master's thesis, Queen's Univ., 2008. 25, 26
- [26] Jimpeng Li, Harold Mouchère, and Christian Viard-Gaudin, *Reducing Annotation Workload Using a Codebook Mapping and its Evaluation in On-Line Handwriting*, Proceedings of 2012 International Conference on Frontiers in Handwriting Recognition (ICFHR-2012) (Bari, Italie), 2012, pp. 1–6 (Anglais). ix, 2, 29, 30
- [27] Scott MacLean, George Labahn, Edward Lank, Mirette Marzouk, and David Tausky, *Grammar-based techniques for creating ground-truthed sketch corpora*, Int. J. Doc. Anal. Recognit. **14** (2011), 65–74. ix, 24, 31
- [28] Scott MacLean, David Tausky, George Labahn, Edward Lank, and Mirette Marzouk, *Tools for the efficient generation of hand-drawn corpora based on context-free grammars*, Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling (New York, NY, USA), SBIM '09, ACM, 2009, pp. 125–132. 25
- [29] Nicholas E. Matsakis, *Recognition of handwritten mathematical expressions*, Master's thesis, Massachusetts Institute of Technology, 1999. 1
- [30] Adiel Mittmann, *Reconhecimento de padrões*, 2007, <http://www.inf.ufsc.br/~patrec/intro.html>, [Online; acesso 07 de abril de 2012]. ix, 13
- [31] Harold Mouchere, Christian Viard-Gaudin, Dae Hwan Kim, Jin Hyung Kim, and Utpal Garain, *Crohme2011: Competition on recognition of online handwritten mathematical expressions*, Proceedings of the 2011 International Conference on Document Analysis and Recognition (Washington, DC, USA), ICDAR '11, IEEE Computer Society, 2011, pp. 1497–1500. 2
- [32] J. Munkres, P. Foggia, C. Sansone, and M Vento, *Algorithms for the assignment and transportation problems*, J. Soc. Indust. Appl. Math. **18** (1957), 32–38. 9
- [33] Alexandre Noma, Alvaro Pardo, and Roberto M. Cesar, Jr., *Structural matching of 2d electrophoresis gels using deformed graphs*, Pattern Recogn. Lett. **32** (2011), 3–11. 14, 39
- [34] R. Plamondon and S. N. Srihari, *On-line and off-line handwriting recognition: A comprehensive survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000), no. 1, 63–84. 19
- [35] L. Prasanth, V. Babu, R. Sharma, G. V. Rao, and Dinesh M., *Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features*, Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02, 2007, pp. 1028–1032. 45
- [36] Solen Quiniou, Harold Mouchere, Sebastián Pen Saldarriaga, Christian Viard-Gaudin, Emmanuel Morin, Simon Petitrenaud, and Sofiane Medjkoune, *Hamex - a handwritten and audio dataset of mathematical expressions*, Proceedings of the 2011 International Conference on Document Analysis and Recognition (Washington, DC, USA), ICDAR '11, IEEE Computer Society, 2011, pp. 452–456. 26, 27
- [37] Dmitri A. Rachkovskij and Ernst M. Kussul, *Datagen: a generator of datasets for evaluation of classification algorithms*, Pattern Recogn. Lett. **19** (1998), 537–544. 26
- [38] Taik Heon Rhee and Jin Hyung Kim, *Efficient search strategy in structural analysis for handwritten mathematical expression recognition*, Pattern Recogn. **42** (2009), 3192–3201. 25, 27

- [39] Kunal Sain, Abhishek Dasgupta, and Utpal Garain, *Emers: a tree matching-based performance evaluation of mathematical expression recognition systems*, Int. J. Doc. Anal. Recognit. **14** (2011), no. 1, 75–85. [27](#)
- [40] Yu SHI, HaiYang LI, and Frank K. SOONG, *A unified framework for symbol segmentation and recognition of handwritten mathematical expressions*, Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02 (Washington, DC, USA), IEEE Computer Society, 2007, pp. 854–858. [21](#)
- [41] Ricardo Sider, B. Y. Ozahata, and Nina S. T. Hirata, *Expressmath: análise estrutural de expressões matemáticas manuscritas.*, Iniciação científica e tecnológica: o jovem pesquisador em ação II., 2010, pp. 109–129. [41](#)
- [42] Elena Smirnova and Stephen M. Watt, *Communicating mathematics via pen-based interfaces*, Symbolic and Numeric Algorithms for Scientific Computing, International Symposium on **0** (2008), 9–18. [1](#), [30](#)
- [43] Ernesto Tapia, *Understanding mathematics: A system for the recognition of on-line handwritten mathematical expression*, Ph.D. thesis, Freie Universität Berlin, 2004. [22](#)
- [44] Ernesto Tapia and Raul Rojas, *Recognition of on-line handwritten mathematical expressions in the e-chalk system - an extension*, Document Analysis and Recognition, International Conference on **0** (2005), 1206–1210. [20](#), [22](#), [24](#)
- [45] Mariano Tepper, Daniel Acevedo, Norberto Goussies, Julio Jacobo, and Marta Mejail, *A decision step for shape context matching*, Proceedings of the 16th IEEE international conference on Image processing (Piscataway, NJ, USA), ICIP'09, IEEE Press, 2009, pp. 409–412. [ix](#), [16](#)
- [46] Ana Paula Tomás, *Emparelhamentos, casamentos estáveis e algoritmos de colocação de professores*, Tech. Report DCC-2005-02, Universidade do Porto, Rua do Campo Alegre, 823, 4150 Porto, Portugal, March 2005. [9](#)
- [47] J.-Y. Toumit, S. Garcia-Salicetti, and H. Emptoz, *A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents*, Proceedings of the Fifth International Conference on Document Analysis and Recognition (Washington, DC, USA), ICDAR '99, IEEE Computer Society, 1999, pp. 119–. [1](#)
- [48] Seiichi Toyota, Seiichi Uchida, and Masakazu Suzuki, *Structural analysis of mathematical formulae with verification based on formula description grammar*, Proceedings of the 7th international conference on Document Analysis Systems (Berlin, Heidelberg), DAS'06, Springer-Verlag, 2006, pp. 153–163. [24](#)
- [49] Seiichi Uchida and Hiroaki Sakoe, *Handwritten character recognition using elastic matching based on a class-dependent deformation model*, Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 1 (Washington, DC, USA), ICDAR '03, IEEE Computer Society, 2003, pp. 163–. [26](#)
- [50] Ba-Quy Vuong, Siu-Cheung Hui, and Yulan He, *Progressive structural analysis for dynamic recognition of on-line handwritten mathematical expressions*, Pattern Recogn. Lett. **29** (2008), 647–655. [ix](#), [24](#)
- [51] Xiaofang Xie, *On the recognition of handwritten mathematical symbols*, Ph.D. thesis, The University of Western Ontario, London, Ontario, December 2007. [ix](#), [1](#), [19](#), [22](#), [23](#), [26](#), [66](#)

- [52] Ryo Yamamoto, Shinji Sako, Takuya Nishimoto, and Shigeki Sagayama, *On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar*, International Workshop on Frontiers in Handwriting Recognition, 2006. [23](#)
- [53] Richard Zanibbi, Dorothea Blostein, and James R. Cordy, *Baseline structure analysis of handwritten mathematics notation*, Document Analysis and Recognition, International Conference on **0** (2001), 0768. [66](#)
- [54] Richard Zanibbi, Dorothea Blostein, and James R. Cordy, *Recognizing mathematical expressions using tree transformation*, IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002), 1455–1467. [22](#), [26](#)
- [55] ———, *Recognizing mathematical expressions using tree transformation*, IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002), no. 11, 1455–1467. [24](#)
- [56] Dong-Qing Zhang and Shih-Fu Chang, *Detecting image near-duplicate by stochastic attributed relational graph matching with learning*, Proceedings of the 12th annual ACM international conference on Multimedia (New York, NY, USA), MULTIMEDIA '04, ACM, 2004, pp. 877–884. [5](#)
- [57] Daniel Zwillinger, *Standard mathematical tables and formulae*, 31st ed., Chapman and Hall/-CRC, 2003. [43](#)