

Projeto de Leiaute Semi-Automático utilizando Dispositivo Adaptativo

P. R. M. Cereda

Resumo—Este artigo apresenta um estudo preliminar sobre a utilização de dispositivos adaptativos no processo de elaboração de leiautes. Este caso específico pode ser generalizado para o problema de particionamento de regiões. Durante a fase de projeto, foi utilizado um modelo semi-automático para geração de um leiaute simplificado baseado em um estudo de caso.

Palavras-chave:—Editoração de texto, Diagramação, Dispositivo adaptativo, Particionamento de regiões

I. INTRODUÇÃO

A diagramação de texto consiste no processo de distribuir elementos textuais na área disponível de uma página. Os textos podem compartilhar seu espaço com outros elementos gráficos, tais como imagens e formas. A diagramação é uma etapa primordial na publicação de materiais gráficos, tais como jornais, revistas e periódicos [1].

O profissional designado para a tarefa de diagramação deve levar em consideração uma série de critérios para realizar a distribuição dos elementos em uma página, tais como tamanho dos elementos e nível hierárquico. A diagramação costuma ser um processo oneroso em tempo e esforço.

Este artigo procura realizar um estudo preliminar sobre a utilização de dispositivos adaptativos no processo de diagramação, na tentativa de torná-lo mais fluente, simplificado e proporcionar meios para predição de espaços sub-ótimos para a acomodação dos elementos textuais e gráficos. Uma visualização mais abstrata do processo de diagramação pode ser generalizada para o problema de particionamento de regiões, tema recorrente em diversas áreas de pesquisa, incluindo Inteligência Artificial e Pesquisa Operacional.

A organização deste artigo é a seguinte: na Seção II, o processo de diagramação é apresentado, abrangendo suas características principais. Na Seção III, o projeto de leiaute semi-automático é contextualizado. A Seção IV apresenta uma implementação do modelo proposto. Os experimentos e análises são apresentados na Seção V. As considerações finais são apresentadas na Seção VI.

II. DIAGRAMAÇÃO

O processo de diagramação define um conjunto de categorias de elementos que determinam seus níveis hierárquicos e aspectos tipográficos em uma disposição de página. Além disso, existem critérios específicos para cada tipo de publicação, nos quais os elementos podem receber regras específicas que têm prioridade sobre outras regras. Alguns dos elementos textuais incluem títulos, subtítulos, corpos de texto, epígrafes,

notas de rodapé, caixas de texto e blocos de citações. Cada um destes elementos possui características de formatação e disposição hierárquica distintas [1].

Para que os elementos textuais sejam incorporados na área disponível de uma página, é necessário que exista um projeto de leiaute, isto é, um conjunto de diretrizes e critérios que determine a disposição. Algumas dessas diretrizes incluem número de colunas, margens de página, espaçamento entre elementos, hifenização, alinhamento textual e orientação. A Figura 1 apresenta um exemplo de leiaute de página contendo alguns elementos textuais e gráficos, a saber: (1) dimensão total da página, (2) corpo de texto, (3) figura inserida ao lado do texto, (4) linha divisória, e (5) margens da página.

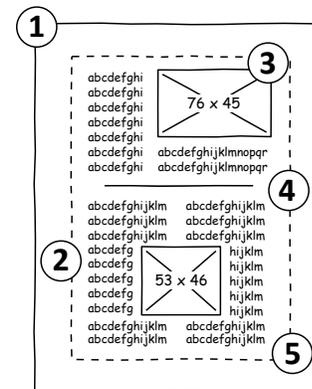


Figura 1. Exemplo de leiaute de página, contendo alguns elementos textuais e gráficos. A área tracejada denota as margens da página.

Um projeto de leiaute deve também levar em consideração aspectos tipográficos e de ergonomia. Tamanhos de fontes pequenos ou famílias de fontes com traços irregulares podem comprometer a visão do leitor, dificultando a leitura e a própria compreensão do texto. É também interessante que respeite-se, sempre que possível, o fluxo natural de um texto – nos países ocidentais, de cima para baixo, da esquerda para a direita – e que linhas orfãs ou parágrafos curtos não sejam isolados em blocos distintos.

III. PROJETO DE LEIAUTE SEMI-AUTOMÁTICO

A área de inteligência artificial apresenta vários estudos para oferecer soluções automáticas para o problema de particionamento de regiões e, em particular, na resolução da disposição hierárquica de elementos textuais e gráficos em uma página. Os modelos propostos apresentam resultados interessantes, inclusive na avaliação de leiautes com um alto nível de complexidade [2], [3], [4], [5], [6].

Apesar de extremamente eficientes, os modelos tradicionais de projeto de leiaute automático requerem um conjunto de regras imutáveis, o que limita a abrangência do algoritmo e dificulta sua extensão para outras classes, além de demandarem tempo e esforço computacionais consideráveis [3]. O processo de diagramação exige rapidez e agilidade na disposição dos elementos na página por parte do profissional; algoritmos eficientes mas onerosos em tempo e recursos podem comprometer seriamente o fluxo de trabalho dos corpos editoriais.

Com o advento da Internet e popularização das redes sociais, os conteúdos de jornais, revistas e periódicos passaram a ser disponibilizados em vários formatos. O processo de diagramação sofreu uma transformação para contemplar vários projetos de leiaute em um mesmo documento, de acordo com cada formato de distribuição, de modo simultâneo - meio eletrônico, impressão monocromática ou colorida, conteúdo para dispositivos móveis, livros digitais, entre outros. A necessidade de automação do processo de leiaute tornou-se muito mais evidente, principalmente em grandes veículos de informação [3].

A tecnologia adaptativa tem obtido resultados interessantes na utilização de dispositivos adaptativos aplicados em problemas tradicionalmente tratados com técnicas de inteligência artificial [7]. A simplicidade e o poder computacional de tais dispositivos contribuem para sua ampla utilização, além de serem eficientes em tempo e espaço [8]. Este artigo propõe um modelo computacional utilizando um dispositivo adaptativo para construir um projeto de leiaute semi-automático, ao invés das técnicas tradicionais de inteligência artificial geralmente empregadas nesta particular classe do problema de particionamento de regiões.

O modelo proposto é dito *semi-automático* porque é esperado que o diagramador forneça sua ordem inicial preferida para os elementos textuais a serem inseridos na página. O dispositivo adaptativo então tentará manter-se o mais fiel possível a essa ordem.

O leiaute escolhido para o modelo proposto é o mais simplificado possível: respeitando as margens da página, o dispositivo adaptativo tentará dispor uma quantidade arbitrária de blocos de texto independentes entre si em duas colunas não-balanceadas. Os blocos de textos podem continuar em outras páginas, caso seja necessário. A Figura 2 apresenta um esboço do leiaute a ser obtido.

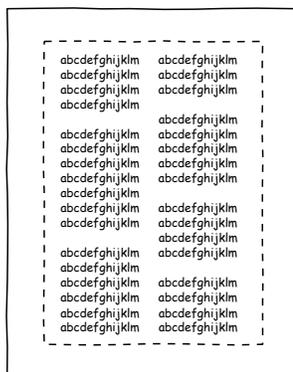


Figura 2. Esboço do leiaute a ser obtido através do modelo proposto.

Algumas diretrizes foram impostas ao modelo para representar o contexto de um projeto de leiaute real. Dois grupos de leiaute foram definidos no escopo deste artigo, cada um com suas próprias diretrizes, apresentados a seguir.

O primeiro grupo consiste em um leiaute para blocos de textos com eventual troca de posição entre blocos, sem quebra de texto para a página seguinte. As diretrizes definidas são:

- 1) A troca de posição entre blocos de textos é permitida, mas o dispositivo adaptativo tentará manter-se o mais fiel possível à ordem inicial.
- 2) Um bloco de texto pode iniciar em uma coluna e continuar em outra, mas não pode iniciar em uma página e continuar em outra.

O segundo grupo contempla um leiaute para blocos de textos respeitando a ordem dos blocos, com eventuais repetições de partes dos textos, de acordo com o contexto e disposição. As diretrizes definidas são:

- 1) A troca de posição entre blocos de textos é proibida, de forma que a ordem dos elementos seja sempre preservada.
- 2) O dispositivo adaptativo poderá repetir partes do texto de acordo com uma semântica estabelecida previamente. Por exemplo, refrões podem ser repetidos quando o bloco de texto continuar na página seguinte.
- 3) Um bloco de texto pode fluir por todo o leiaute da página, inclusive continuando em outra página, se necessário. O dispositivo adaptativo deve apenas evitar linhas órfãs e deslocamento de parágrafos curtos.

A Figura 3 apresenta um exemplo de leiaute real com diretrizes semelhantes às do segundo grupo. Observe que o bloco em negrito é repetido na página seguinte para facilitar a leitura. O leiaute em questão é proveniente do semanário litúrgico *Deus Conscio*, uma publicação da Editora Santuário contendo o ordinário de missa do dia, e amplamente utilizado por algumas comunidades católicas para acompanhar as missas.

<p>para ela com agrado e estendeu-lhe o cetro de ouro que tinha na mão, e Ester aproximou-se para tocar a ponta do cetro.</p> <p>^{72b}Então, o rei lhe disse: "O que me pedes, Ester, o que queres que eu faça? Ainda que me pedisses a metade do meu reino, ela te seria concedida".</p> <p>Ester respondeu-lhe: "Se ganhei as tuas boas graças, ó rei, e se for de teu agrado, concede-me a vida — eis o meu pedido! — e a vida do meu povo — eis o meu desejo!" — Palavra do Senhor.</p> <p>Ass.: Graças a Deus!</p>	<p>— Escutai, minha filha, olhai, ouvi isto! "Esqueci vosso povo e a casa paterna! Que o Rei se encante com vossa beleza! Prestai-lhe homenagem: é vosso Senhor!</p> <p>Ass.: Escutai, minha filha, olhai, ouvi isto:/ que o Rei se encante com vossa beleza!</p> <p>— O povo de Tiro vos traz seus presentes, / os grandes do povo vos pedem favores / Majestosa, a princesa real vem chegando / vestida de ricos brocados de ouro.</p> <p>— Em vestes vistosas ao Rei se dirige / e as virgens amigas lhe formam cortejo / entre cantos de festa e com grande alegria / ingressam, então, no palácio real".</p>
<p>7. Salmo Responsorial (Sl 44)</p> <p>Salmista: Escutai, minha filha, olhai, ouvi isto / que o Rei se encante com vossa beleza!</p> <p>Ass.: Escutai, minha filha, olhai, ouvi isto / que o Rei se encante com vossa beleza!</p>	<p>8. Segunda Leitura (Ap 12.1.5.13a.15-16a)</p> <p>Livro do Apocalipse de São João:</p> <p>"Apareceu no céu um grande sinal; uma mulher vestida do sol, tendo a lua debaixo dos pés e so-</p>

Figura 3. Exemplo de leiaute real com diretrizes semelhantes às do segundo grupo. O leiaute em questão é proveniente do semanário litúrgico *Deus Conscio*.

A Tabela I resume os grupos de leiaute utilizados para a definição do modelo de projeto de leiaute semi-automático e suas diretrizes.

O dispositivo adaptativo escolhido para representar o modelo de projeto de leiaute proposto foi o autômato adaptativo, devido à sua simplicidade e poder computacional [9], [10]. É importante destacar que outros dispositivos adaptativos poderiam ser utilizados neste caso; o autômato foi escolhido por uma preferência do autor.

Tabela I
GRUPOS DE LEIAUTE DEFINIDOS NO ESCOPO DO ARTIGO.

	Grupo 1	Grupo 2
Troca de posições	Sim	Não
Respeitar ordem	Se possível	Sim
Fluir em outra página	Não	Sim
Blocos como átomos	Sim	Não
Partes convencionais	Não	Sim
Partes de repetição	Não	Sim

A primeira etapa da definição do modelo é realizar a divisão do espaço disponível de uma página em regiões menores. Para o modelo proposto, optou-se pela divisão em oito regiões de dimensões iguais, com quatro regiões por coluna, conforme ilustrado na Figura 4. Considerando as dimensões das margens como $M = a \times b$, onde a denota a altura e b a largura, cada região i terá as dimensões $R_i = c \times d$, onde $c = \frac{a}{4}$ e $d = \frac{b}{2}$.

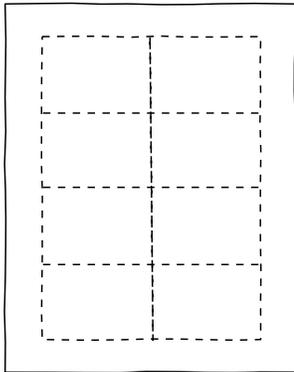


Figura 4. Esboço do leiaute a ser obtido através do modelo proposto.

A etapa seguinte consiste em discretizar os blocos de texto a serem inseridos em termos das regiões calculadas no passo anterior. Cada bloco de texto i a ser inserido no espaço disponível da página terá sua área A_i calculada e depois comparada com a área da região, $A_{\text{região}} = cd$, para determinar o número de regiões que o bloco i ocupará. O cálculo do número de regiões N_i é apresentado na Fórmula 1.

$$N_i = \left\lceil \frac{A_i}{A_{\text{região}}} \right\rceil \quad (1)$$

É importante salientar que o modelo definido neste artigo não suporta valores de N_i maiores do que oito regiões, isto é, textos que extrapolam uma página inteira. Esta decisão foi tomada por questões de simplicidade.

O autômato adaptativo é modelado de acordo com os blocos de textos disponíveis. Cada bloco de texto será representado por um estado q_i do autômato. O alfabeto de entrada Σ adota as letras do alfabeto latino em ordem crescente denotando o número de regiões de cada bloco, isto é, todos símbolos e seus respectivos significados apresentados na Tabela II estarão disponíveis.

A modelagem a seguir contempla o primeiro grupo de leiaute. Cada transição do automato adaptativo do modelo consome um símbolo $\sigma_i \in \Sigma$ que denota a quantidade de regiões

Tabela II
SÍMBOLOS DO ALFABETO DE ENTRADA Σ E SEUS REPECTIVOS SIGNIFICADOS NO PRIMEIRO GRUPO DE LEIAUTE.

Símbolo	N_i	Símbolo	N_i
a	1	e	5
b	2	f	6
c	3	g	7
d	4	h	8

ocupadas pelo bloco de código seguinte – o estado de destino da transição corrente. As funções adaptativas do autômato mapeiam as diretrizes do modelo de projeto de leiaute; caso um bloco de texto apresente uma inconsistência de leiaute, provavelmente devido ao número de regiões necessário para sua correta inclusão, a função adaptativa associada à transição corrente realiza uma modificação na topologia do autômato, alterando as transições seguintes ou inserindo estados novos.

A criação de um estado novo no conjunto de estados disponíveis Q indica que o modelo de projeto de leiaute não pôde encontrar um bloco de texto que tenha o número mínimo de regiões para ser inserido na posição corrente. Para não comprometer a diagramação dos elementos subsequentes, o autômato adiciona um estado preenchendo a região corrente tantas vezes quanto for necessário até que um bloco de texto tenha o número de regiões válidas para então ser inserido.

O autômato adaptativo do modelo tem uma execução extremamente compacta. As funções adaptativas são executadas de acordo com o símbolo corrente da cadeia de entrada e seu respectivo significado. Por exemplo, se existem apenas dois blocos disponíveis na página e o símbolo corrente denota um número de regiões maior do que esse valor, a função adaptativa será executada e tentará encontrar um bloco de texto que atenda ao requisito; caso não existam blocos de texto com o número mínimo de regiões, um novo estado denotando um bloco de texto *dummy* é então inserido, e o reconhecimento do restante da cadeia prossegue.

Como exemplo, considere oito blocos de textos já discretizados para serem avaliados pelo autômato adaptativo. A Tabela III apresenta o número de regiões de cada bloco.

Tabela III
NÚMERO DE REGIÕES DE CADA BLOCO.

Bloco	N_i	Bloco	N_i
1	2	5	1
2	3	6	2
3	2	7	2
4	2	8	2

A partir do número de regiões de cada bloco e de sua ordem estabelecida, e de acordo com a Tabela II, a cadeia a ser submetida ao autômato terá a forma $\langle bcbabb \rangle$. Uma representação do autômato adaptativo do primeiro grupo de leiaute é apresentado na Figura 5, transposto no leiaute das páginas, para fácil visualização. As transições foram omitidas, e as cores preenchem a quantidade de regiões ocupadas por cada bloco, denotado por cada estado.

É possível notar que, de acordo com a Figura 5, existe

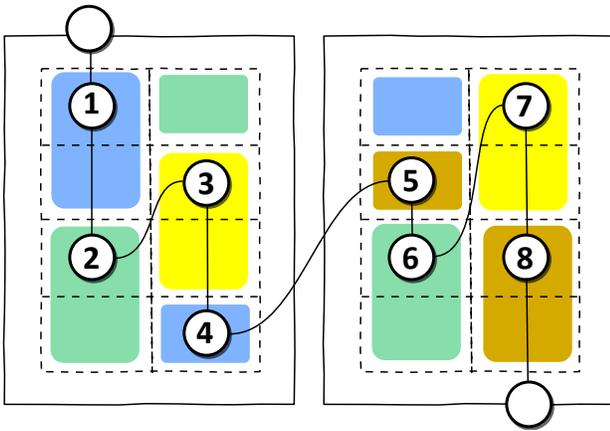


Figura 5. Representação do autômato adaptativo do primeiro grupo de leiaute.

um bloco de texto violando uma das diretrizes do primeiro grupo de leiaute para o modelo – o bloco número quatro está iniciando o texto na primeira página e encerrando na segunda. Para que a diretriz sobre praticidade e ergonomia seja contemplada, o autômato adaptativo tentará, por meio das funções adaptativas, redefinir a posição do bloco de texto problemático.

Uma possível disposição dos oito blocos de textos nas duas páginas, sem violar as diretrizes e tentando manter-se o mais fiel possível à ordem dos elementos, é apresentada na Figura 6. É importante observar que essa ilustração apresenta, de modo simplificado, a lógica de como as funções adaptativas atuarão sobre os estados e transições.

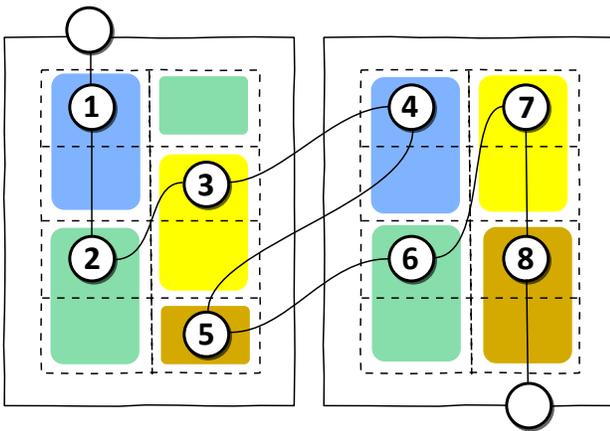


Figura 6. Possível disposição dos oito blocos de textos nas duas páginas, sem violar as diretrizes.

O autômato adaptativo resultante após o reconhecimento da cadeia $\langle bebbabbb \rangle$ é apresentado na Figura 7. Note que as funções adaptativas removeram as transições $q_3 \rightarrow q_4$, $q_4 \rightarrow q_5$ e $q_5 \rightarrow q_6$, e as novas transições $q_3 \rightarrow q_5$, $q_5 \rightarrow q_4$ e $q_4 \rightarrow q_6$ foram inseridas. A configuração final do autômato – ou melhor, a ordem de seus estados – determina a disposição dos blocos de textos nas páginas.

As funções adaptativas atuam sobre a disponibilidade de regiões na página corrente, reconfigurando os blocos de textos para compôr uma região válida, ou simplesmente marcar

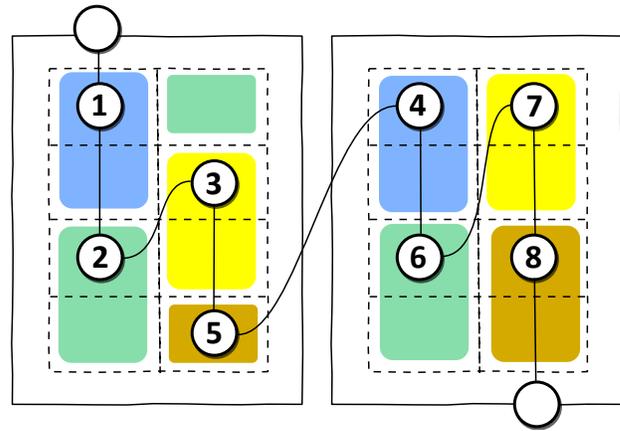


Figura 7. Representação da configuração final do autômato adaptativo do primeiro grupo de leiaute.

o espaço como insuficiente, através da inserção de novos estados.

A modelagem do autômato adaptativo do segundo grupo de leiaute é semelhante à do primeiro grupo, com a exceção do tratamento dos blocos de textos. No primeiro grupo, os blocos de textos eram tratados como elementos atômicos; no segundo grupo, existe uma subdivisão dentro do próprio bloco de texto. Para simplificar a compreensão, a subdivisão foi definida como o número de regiões N_i (Fórmula 1) do bloco de texto i , isto é, um bloco de texto i com $N_i = 4$ terá quatro subdivisões (ou partes) dentro do bloco.

Uma parte do bloco de texto pode admitir repetição. Por exemplo, suponha que um canto tenha três estrofes e um refrão (estrofe principal); o último pode ser repetido entre cada estrofe. Um jogral pode também admitir repetições de partes. Por uma questão de ergonomia, a repetição pode ser útil em alguns casos – se o refrão encontra-se em uma página e as estrofes seguintes em outra, é necessário alternar páginas para acompanhar a leitura; a repetição do refrão na nova página poupa esforços e facilita o manuseio e compreensão.

O alfabeto de entrada Σ , no segundo grupo de leiaute, adota as letras do alfabeto latino, no qual cada letra denota uma parte determinada. Letras iguais referem-se a um bloco de texto. O tamanho desta subcadeia denota o número de regiões do bloco de texto; por exemplo, $\langle aaa \rangle$ denota o bloco de texto a com o número de regiões $N_a = |aaa| = 3$.

As letras minúsculas denotam partes convencionais do bloco de texto; analogamente, as letras maiúsculas referem-se às partes de repetição. Por exemplo, $\langle bBbb \rangle$ pode denotar um bloco de texto com três estrofes e um refrão.

Como exemplo, considere a cadeia de entrada na forma $\langle aABbCcDddeEFfGg \rangle$ a ser submetida ao autômato, denotando sete blocos de textos, cada qual com suas partes convencionais e de repetição. Uma representação do autômato adaptativo do segundo grupo de leiaute é apresentado na Figura 8, transposto no leiaute das páginas, para fácil visualização. As transições foram omitidas, e as cores preenchem a quantidade de regiões ocupadas por cada bloco, denotado por cada estado. As marcações listradas denotam as partes de repetições.

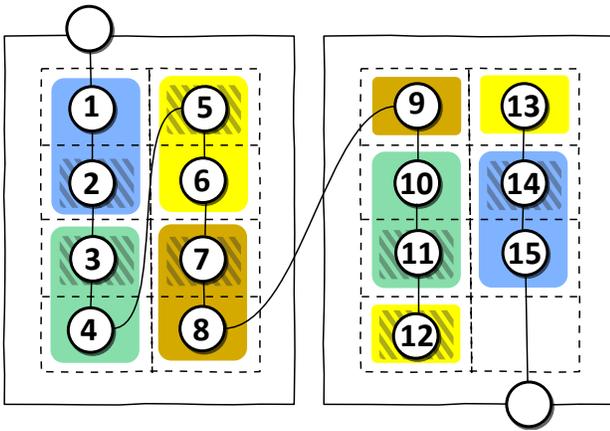


Figura 8. Representação do autômato adaptativo do segundo grupo de leiaute.

É possível notar que, de acordo com a Figura 8, existe um bloco de texto violando uma das diretrizes do segundo grupo de leiaute para o modelo – o bloco número quatro inicia em uma página, com uma parte de repetição, e termina em outra página, com apenas uma parte convencional. Para que a diretriz sobre ergonomia seja contemplada, o autômato adaptativo tentará, por meio das funções adaptativas, reproduzir a parte de repetição na segunda página.

O autômato adaptativo resultante após o reconhecimento da cadeia $\langle aABbCcDddeEFfGg \rangle$ é apresentado na Figura 9. Note que as funções adaptativas removeram a transição $q_8 \rightarrow q_9$ que liga duas partes convencionais, criaram um novo estado q_{16} idêntico a q_7 – uma parte de repetição – e as novas transições $q_8 \rightarrow q_{16}$ e $q_{16} \rightarrow q_9$ foram inseridas. A configuração final do autômato – ou melhor, a ordem de seus estados – determina a disposição das partes dos blocos de textos nas páginas.

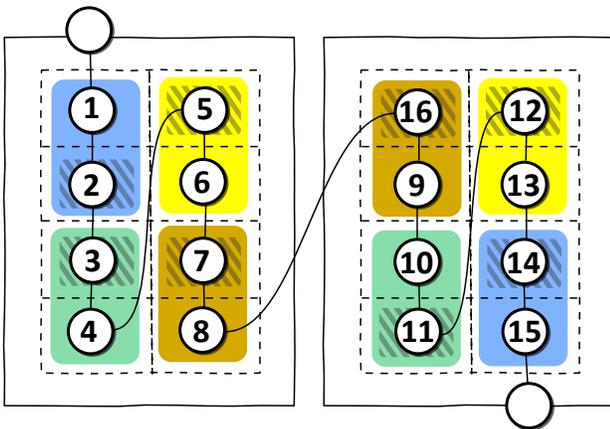


Figura 9. Representação da configuração final do autômato adaptativo do segundo grupo de leiaute.

As funções adaptativas atuam sobre a distribuição semântica das partes de um bloco de texto na página corrente, reproduzindo partes de repetição entre partes convencionais para compor um bloco de texto válido, ou simplesmente evitar linhas órfãs e parágrafos curtos.

O autômato adaptativo do modelo pode representar várias

classes de projetos de leiaute, incluindo novas diretrizes e critérios hierárquicos de inserção. A página também pode ser subdividida e ter autômatos independentes atuando em cada partição, com regras e restrições exclusivos de cada área.

IV. IMPLEMENTAÇÃO

O modelo do projeto de leiaute semi-automático apresentado na Seção III foi implementado utilizando a linguagem Python e executado em um ambiente Linux de 64 bits. O modelo foi disponibilizado como um módulo, isto é, um arquivo contendo definições e declarações da linguagem. A utilização do módulo requer apenas sua importação, através do comando:

```
>>> import aalayout
```

Na fase de definição dos aspectos de implementação, optou-se por traduzir as diretrizes de projeto de leiaute diretamente no código-fonte. Estuda-se a possibilidade de torná-las independentes, escritas em uma linguagem de marcação – por exemplo, XML – e carregadas em tempo de execução pelo módulo de leiaute. Por ora, apenas as diretrizes dos dois grupos de leiaute apresentados na Seção III estão disponíveis.

A utilização do módulo é semelhante ao processo de reconhecimento de uma cadeia w pelo autômato adaptativo do modelo; é suficiente fornecer a cadeia de entrada e o identificador do grupo de leiaute escolhido:

```
>>> aalayout.generateLayout('bcbbabbb', 1)
- Input:
[1, 2, 3, 4, 5, 6, 7, 8]
- Output:
[[1, 2, 3, 5], [4, 6, 7, 8]]
```

A saída da função `generateLayout` é uma lista de listas. Cada lista interna representa uma página, com seus respectivos blocos de textos. Caso existam blocos de textos *dummy*, eles serão representados pela letra X, indicando que não existiam blocos de texto com o número mínimo de regiões para concluir a diagramação da página corrente:

```
>>> aalayout.generateLayout('cccc', 1)
- Input:
[1, 2, 3, 4]
- Output:
[[1, 2, X], [3, 4, X]]
```

Quando o segundo grupo de leiaute é escolhido, a saída da função `generateLayout` é ligeiramente diferente:

```
>>> aalayout.generateLayout('aABbCcDddeEFfGg', 2)
- Input:
[[1, 2], [3, 4], [5, 6], [7, 8, 9], [10, 11], [12, 13], [14, 15]]
- Output:
[[[1, 2], [3, 4], [5, 6], [7, 8]], [7, 9], [10, 11], [12, 13], [14, 15]]]
```

Neste caso, a lista mais interna representa blocos de textos, com suas respectivas partes convencionais e de repetição. A próxima lista no nível superior representa uma página, contendo os blocos de textos. Caso existam partes a serem repetidas, seu identificador é repetido nas regiões correspondentes.

O módulo é extremamente compacto e compatível com as séries 2 e 3 da linguagem Python. É possível utilizá-lo autonomamente, na forma de script, ou incluí-lo em programas já existentes.

V. EXPERIMENTO E ANÁLISE

Foi realizado um experimento semelhante aos exemplos sintéticos apresentados na Seção III para confirmar os resultados obtidos. O experimento consistiu na diagramação de dois folhetos de cantos religiosos de uma comunidade católica do interior do estado de São Paulo. O primeiro folheto consistia em oito cantos, dispostos em duas colunas, ao longo de duas páginas no formato A5. As diretrizes do projeto de leiaute utilizadas coincidiram com as do primeiro grupo de leiaute. O segundo folheto consistia em sete cantos, dispostos em duas colunas, ao longo de duas páginas no formato A5. As diretrizes do projeto de leiaute utilizadas coincidiram com as do segundo grupo de leiaute.

O experimento foi iniciado com a diagramação do primeiro folheto. Na primeira etapa, os cantos foram discretizados para determinar suas dimensões em cada região disponível da página, considerando as margens. A classificação dos oito cantos do primeiro folheto é apresentada na Tabela IV.

Tabela IV

NÚMERO DE REGIÕES PARA CADA CANTO DO PRIMEIRO FOLHETO DO EXPERIMENTO.

Bloco	N_i	Bloco	N_i
1	2	5	1
2	3	6	2
3	2	7	2
4	2	8	2

Como comparação, a Figura 10 apresenta uma diagramação linear do primeiro folheto sem levar em conta as diretrizes do projeto de leiaute. É possível observar a quebra do quarto canto para a próxima página – os traços horizontais no folheto indicam a separação entre blocos de cantos.

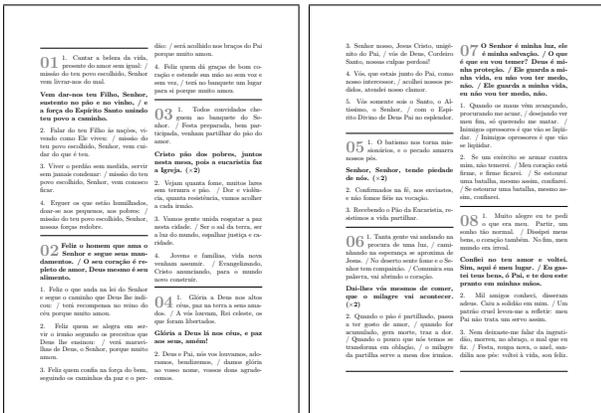


Figura 10. Diagramação linear do primeiro folheto, sem diretrizes.

De acordo com a Tabela IV, a cadeia a ser submetida ao autômato adaptativo também apresentou a forma $\langle bcbabb \rangle$. A topologia inicial do autômato é praticamente idêntica à representação do autômato da Figura 5.

Após a submissão da cadeia $\langle bcbabb \rangle$, o autômato resultante apresenta a disposição desejada dos cantos no folheto, conforme ilustrado na Figura 11. A topologia final do autômato também é idêntica à representação do autômato da Figura 7.

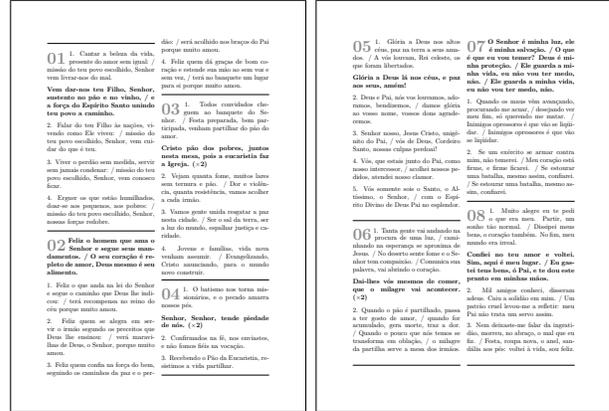


Figura 11. Diagramação final do primeiro folheto, de acordo com a topologia do autômato adaptativo resultante do modelo do primeiro grupo de leiaute.

A segunda parte do experimento consistiu na diagramação do segundo folheto, utilizando as diretrizes do segundo grupo de leiaute. Na primeira etapa, os cantos foram discretizados para determinar suas dimensões em cada região disponível da página, considerando as margens. A classificação dos sete cantos do segundo folheto é apresentada na Tabela V.

Tabela V

NÚMERO DE REGIÕES PARA CADA CANTO DO SEGUNDO FOLHETO DO EXPERIMENTO.

Bloco	N_i	Bloco	N_i
1	2	5	2
2	2	6	2
3	2	7	2
4	3		

Como comparação, a Figura 12 apresenta uma diagramação linear do segundo folheto sem levar em conta as diretrizes do projeto de leiaute. É possível observar a quebra do quarto canto para a próxima página, sem a repetição do refrão na nova página.

De acordo com a Tabela V, a cadeia a ser submetida ao autômato adaptativo também apresentou a forma $\langle aABbCcDddeEFfGg \rangle$. A topologia inicial do autômato é praticamente idêntica à representação do autômato da Figura 8.

Após a submissão da cadeia $\langle aABbCcDddeEFfGg \rangle$, o autômato resultante apresenta a disposição desejada dos cantos no folheto, conforme ilustrado na Figura 13. A topologia final do autômato também é idêntica à representação do autômato da Figura 9.

Adicionalmente, foram realizadas avaliações de desempenho com a implementação do modelo do projeto de leiaute semi-automático, apresentada na Seção IV. Os testes consistiram em submeter um conjunto de cantos litúrgicos, devidamente discretizados e classificados, ao módulo de leiaute e calculando seu tempo de execução. O tamanho do conjunto de

- [5] L. M. Pereira, “Artificial intelligence techniques in automatic layout design,” *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, 1978.
- [6] S. M. Casner, “Task-analytic approach to the automated design of graphic presentations,” *ACM Trans. Graph.*, vol. 10, pp. 111–151, 1991.
- [7] J. J. Neto, “Um levantamento da evolução da adaptatividade e da tecnologia adaptativa,” *IEEE Latin America Transactions*, vol. 5, no. 7, pp. 496–505, 2007.
- [8] P. R. M. Cereda and J. J. Neto, “Mineração adaptativa de dados: Aplicação à identificação de indivíduos,” in *WTA 2012: Workshop de Tecnologia Adaptativa*, 2012.
- [9] J. J. Neto, “Contribuições à metodologia de construção de compiladores,” Tese de Livre Docência, Escola Politécnica da Universidade de São Paulo, São Paulo, 1993.
- [10] —, “Adaptive automata for context-dependent languages,” *SIGPLAN Notices*, vol. 29, no. 9, pp. 115–124, 1994.



Paulo Roberto Massa Cereda é graduado em Ciência da Computação pelo Centro Universitário Central Paulista (2005) e mestre em Ciência da Computação pela Universidade Federal de São Carlos (2008). Atualmente, é desenvolvedor de software, atuando em diversos segmentos de mercado, com enfoque principal em software livre. É membro ativo do repositório de código fonte *SourceForge* desde 2005, contribuindo com bibliotecas e softwares de propósito geral. Tem experiência na área de Ciência da Computação, atuando principalmente nos seguintes

temas: inteligência artificial, linguagens de programação, teoria da computação e tecnologia adaptativa.