

Desenvolvimento de documentação para usuário do SCTMF

Willian B. Magalhães¹, Yandre M. e G. da Costa (Orientador)¹

¹Departamento de Informática (DIN) - Universidade Estadual de Maringá (UEM)
Av. Colombo, 5790 - 87020-900 - Maringá - PR - Brasil.

will_magalhaes@yahoo.com.br, yandre@din.uem.br

Resumo. *O software SCTMF (Sistema para Criação e Teste de Modelos Formais) é um sistema multiplataforma, desenvolvido com o intuito de auxiliar no processo do aprendizado de modelos formais estudados em Teoria da Computação. Permite ao usuário criar e avaliar modelos, com base nas informações por ele fornecidas. Este trabalho caracteriza-se pelo estudo dos modelos formais implementados neste software e desenvolvimento de um módulo específico para ajuda ao usuário do SCTMF. Neste módulo o usuário pode encontrar de forma detalhada: explicações sobre os modelos contemplados, auxílio no funcionamento da ferramenta e exemplos práticos para a criação de modelos utilizando o sistema.*

1. Introdução

Dentro da Ciência da Computação existe uma área importante chamada de Teoria da Computação ou TC, nesta área estuda-se tanto fundamentos que descrevem o computador como um modelo matemático, quanto modelos formais que permitem a especificação formal de linguagens. Os modelos estudados em Teoria da Computação podem ser aplicados em diversas áreas, tais como: aplicações de análise léxica e sintática de compiladores, modelagem de circuitos lógicos, sistemas biológicos, entre outros. A Máquina de Turing ou MT, por exemplo, caracteriza-se como a formalização de um procedimento efetivo [Menezes1998]. Pode-se dizer que a MT pode determinar quais problemas são computacionais ou não, ela pode resolver qualquer problema que tenha solução algorítmica.

Conforme [Costa et al. 2008], parte dos alunos possui grande dificuldade em assimilar os modelos formais estudados em Teoria da Computação, já que além de muito abstratos, estes modelos possuem um forte caráter matemático. Dada esta dificuldade em 2007 teve início o desenvolvimento do Sistema para Criação e Teste de Modelos Formais (SCTMF) um ambiente multiplataforma, com o intuito de ajudar no aprendizado de alunos e professores de Teoria da Computação [Meneses e Costa 2007]. A versão mais recente do SCTMF, cujos resultados encontram-se apresentados em [Miranda e Costa 2009], [Bortolotto e Costa 2009] e [Meneguello et al. 2009] disponibilizam módulos que permitem ao usuário criar os oito modelos seguintes: Autômato Finito Determinístico (AFD), Autômato Finito Não Determinístico (AFND), Autômato Finito com Movimento Vazio (AFMV), Expressão Regular (ER), Autômato com Pilha (AP), Gramática Livre de Contexto (GLC), Autômato Linearmente Limitado (ALL) e Máquina de Turing (MT). Também estão disponíveis quatro transformações seguintes: AFND para AFD, minimização de AFD, simplificação de GLC e GLC para FNC. Uma característica importante da ferramenta é a

possibilidade de criar modelos que alcancem todas as classes de linguagem previstas na Hierarquia de Chomsky.

Como a maioria dos softwares existentes o SCTMF também sofre pela falta de documentação de apoio, ou seja, desde o início de seu desenvolvimento em 2007, conta apenas com uma ajuda simplificada contendo apenas informações básicas, sem qualquer detalhamento, explicação ou apresentação de suas funcionalidades. Esta falta de detalhamento implica no aumento da dificuldade na utilização da ferramenta por usuários novos ou com pouco conhecimento dos modelos ou em TC. Este trabalho tem como objetivo fornecer ao usuário do sistema um mecanismo de ajuda completo e detalhado, onde, o usuário consiga compreender o correto funcionamento da ferramenta e de suas funcionalidades, por meio de explicações detalhadas, exemplos práticos e ilustrados sobre a utilização e funcionamento.

Este artigo encontra-se organizado da seguinte forma: na seção 2, serão apresentados os conceitos acerca de Linguagens Formais e Autômatos (LFA), incluindo uma descrição da Hierarquia de Chomsky e dos modelos formais que podem ser criados no ambiente; na seção 3, serão descritos detalhes do ambiente e do seu modo de funcionamento; na seção 4, será apresentado o módulo Help!, seu processo de desenvolvimento e as tecnologias utilizadas neste processo, será apresentado também os conceitos de usabilidade utilizados; na seção 5, serão apresentados as conclusões e os trabalhos futuros.

2. Conceitos de Linguagens Formais e Autômatos (LFA)

Nesta seção e subseções serão apresentados todos os modelos disponíveis na ferramenta SCTMF. Sendo inicialmente apresentada a Hierarquia Chomsky.

2.1. Hierarquia de Chomsky

Esta hierarquia foi estabelecida por Noam Chomsky em 1956, classificando as linguagens em quatro classes hierárquicas sendo: Linguagens Enumeráveis Recursivamente (ou tipo 0), Linguagens Sensíveis ao Contexto (ou tipo 1), Linguagens Livres de Contexto (ou tipo 2) e Linguagens Regulares (ou tipo 3). A imagem abaixo (Figura 1) representa tais classes e como elas relacionam-se entre si.

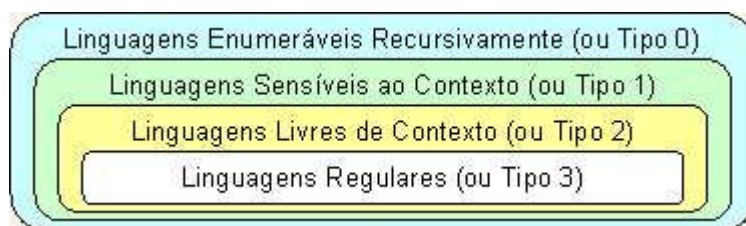


Figura 1. Hierarquia de Chomsky.

Analisando a figura acima pode-se concluir que um modelo capaz de especificar linguagens do tipo 0 seja suficiente para especificar linguagens de qualquer outra classe. No entanto é importante ressaltar que quanto mais abrangente a classe da linguagem maior será a complexidade do formalismo necessário para representá-la, logo para problemas menores e mais específicos torna-se mais vantajoso utilizar um modelo mais simples que simplesmente atenda a necessidade em questão.

2.2. Modelos para Linguagens Regulares

Nesta subseção serão apresentadas as definições matemáticas dos modelos para linguagens regulares disponíveis na ferramenta SCTMF.

O AFD é definido como uma quintupla $\langle \Sigma, S, \delta, S_0, F \rangle$, onde:

- Σ é um alfabeto de símbolos de entrada;
- S é o conjunto finito e não vazio de estados;
- S_0 é o estado inicial, $S_0 \in S$;
- δ é a função de transição de estados, definida $\delta: S \times \Sigma \rightarrow S$;
- F é o conjunto de estados finais, $F \subseteq S$;

O AFND é definido como uma quintupla $\langle \Sigma, S, \delta, S_0, F \rangle$, onde:

- Σ é um alfabeto de símbolos de entrada;
- S é o conjunto finito e não vazio de estados;
- δ é a função de transição, da forma $\delta: S \times \Sigma \rightarrow \rho(S)$;
- S_0 é o conjunto de estados iniciais, finito e não vazio, $S_0 \subseteq S$;
- F é o conjunto de estados finais, $F \subseteq S$.

O AFMV é definido como uma quintupla $\langle \Sigma, S, \delta, S_0, F \rangle$, onde:

- Σ é um alfabeto de símbolos de entrada;
- S é o conjunto finito e não vazio de estados;
- δ é a função de transição, da forma $\delta: S \times (\Sigma \cup \{\lambda\}) \rightarrow \rho(S)$;
- S_0 é o conjunto de estados iniciais, finito e não vazio, $S_0 \subseteq S$;
- F é o conjunto de estados finais, $F \subseteq S$.

Uma Expressão Regular (ER) sobre um alfabeto Σ é definida como segue:

- \emptyset é uma ER e denota a linguagem vazia;
- λ é uma ER e denota a linguagem contendo exclusivamente a palavra vazia $\{\lambda\}$;
- Qualquer símbolo x pertencente a Σ é uma ER e denota a linguagem contendo a palavra x , ou seja, $\{x\}$;
- Seres são ER's e denotam as linguagens R e S , respectivamente, então:
 - $(r+s)$ é ER e denota a linguagem $R \cup S$;
 - (rs) é ER e denota a linguagem $RS = \{uv \mid u \in R \text{ e } v \in S\}$;
 - (r^*) é ER e denota a linguagem R^* .

2.3. Modelos para Linguagens Livres de Contexto

Nesta subseção serão apresentadas as definições matemáticas dos modelos para linguagens livres de contexto disponíveis na ferramenta SCTMF, sendo elas Autômato com Pilha e Gramática Livre de Contexto.

O AP é definido como uma sêxtupla $\langle \Sigma, \Gamma, S, \delta, S_0, B \rangle$, onde:

- Σ é o alfabeto de entrada do AP;
- Γ é o alfabeto da pilha;

- S é o conjunto finito não vazio de estados do AP;
- δ é a função de transição de estados, $\delta: S \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow$ conjunto de subconjuntos finitos de $S \times \Gamma^*$;
- S_0 é o estado inicial, $S_0 \in S$;
- B é o símbolo da base da pilha, $B \in \Gamma$.

A GLC é definida como uma quádrupla $\langle V, T, P, S \rangle$, onde:

- V é um conjunto finito de símbolos não-terminais (ou variáveis);
- T é um conjunto finito de símbolos terminais disjunto de V ;
- P é um conjunto finito de pares, denominados regras de produção tal que a primeira componente é um elemento do conjunto V e a segunda componente é palavra de $(V \cup T)^*$;
- S é um elemento de V , denominado símbolo inicial (ou de partida).

2.4. Modelos para Linguagens Sensíveis ao Contexto

Nesta subsecção serão apresentados os modelos para linguagens sensíveis ao contexto disponíveis na ferramenta.

O ALL é definido como uma ócupla $(\Sigma, S, \delta, S_0, F, V, \langle, \rangle)$, onde:

- Σ é o alfabeto de símbolos de entrada;
- S é o conjunto de estados possíveis, o qual é finito;
- δ é o programa ou função de transição:
 - $\delta: S \times (\Sigma \cup V \cup \{\langle, \rangle\}) \rightarrow 2^S \times (\Sigma \cup V \cup \{\langle, \rangle\}) \times \{E, D\}$ a qual é uma função parcial;
- S_0 é o estado inicial da máquina, tal que S_0 é elemento de S ;
- F é o conjunto de estados finais, tal que F está contido em S ;
- V é o alfabeto auxiliar (pode ser vazio);
- \langle é o símbolo especial marcador de início da fita;
- \rangle é o símbolo especial marcador de final da fita.

2.5. Máquina de Turing

Nesta subsecção serão apresentados os modelos para linguagens sensíveis ao contexto disponíveis na ferramenta.

O MT é definido como uma ócupla $(\Sigma, S, \delta, S_0, F, V, \beta, \langle)$, onde:

- Σ é o alfabeto de símbolos de entrada;
- S é o conjunto de estados possíveis, o qual é finito;
- δ é o programa ou função de transição:
 - $\delta: S \times (\Sigma \cup V \cup \{\beta, \langle\}) \rightarrow S \times (\Sigma \cup V \cup \{\beta, \langle\}) \times \{E, D\}$ a qual é uma função parcial;
- S_0 é o estado inicial da máquina, tal que S_0 é elemento de S ;
- F é o conjunto de estados finais, tal que F está contido em S ;
- V é o alfabeto auxiliar (pode ser vazio);
- β é um símbolo especial para células em branco;

- \langle é o símbolo especial marcador de início da fita.

2.6. Transformações entre modelos

Além dos modelos supracitados a ferramenta SCTMF também conta com quatro tipos de transformações sendo elas: AFND para AFD, minimização de AFD, simplificação de GLC e GLC para FNC.

3. A ferramenta SCTMF

O ambiente SCTMF foi desenvolvido utilizando a linguagem de programação Java, uma tecnologia livre que não depende de plataforma, ou seja, basta que o usuário tenha o interpretador Java (máquina virtual) funcionando em sua máquina para que o sistema funcione independente do sistema operacional que esteja sendo utilizado. A ferramenta SCTMF pode ser acessada pelo endereço www.din.uem.br/yandre/sctmf sob a licença *freeware* (software livre). Atualmente estão disponíveis oito modelos formais (AFD, AFND, AFMV, ER, AP, GLC, ALL e MT) e quatro transformações entre modelos (AFND para AFD, Minimização de AFD, Simplificação de GLC e GLC para FNC).

A Figura 2 mostra a interface principal do sistema contendo na parte superior a barra de menus, nela estão contidas todas as funcionalidades da ferramenta e logo abaixo a barra de modelos onde contém os modelos formais distribuídos de acordo com suas respectivas classes hierárquicas (cada cor corresponde a uma classe hierárquica). O sistema também permite que usuário utilize vários modelos simultaneamente (conforme figura abaixo).

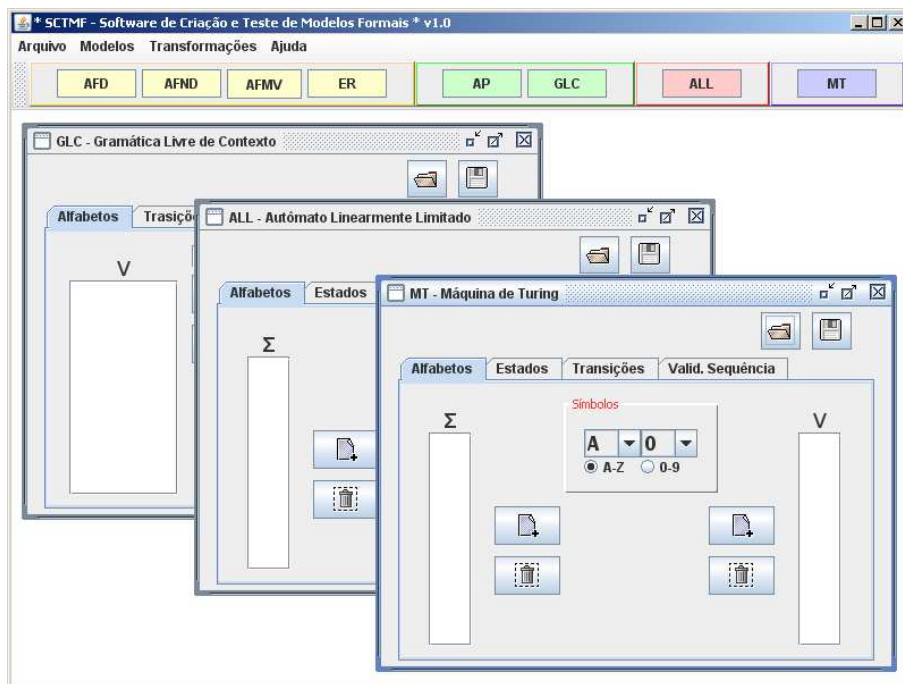


Figura 2. Janela principal SCTMF.

A Figura 3 apresenta um exemplo de interface para criação e teste, neste caso o modelo formal é a Máquina de Turing. Para facilitar o aprendizado da utilização da ferramenta alguns componentes da interface são comuns a todas as outras. Por exemplo, os botões de adicionar e remover, o componente de entrada de símbolos e os botões de salvar modelo e abrir modelo.

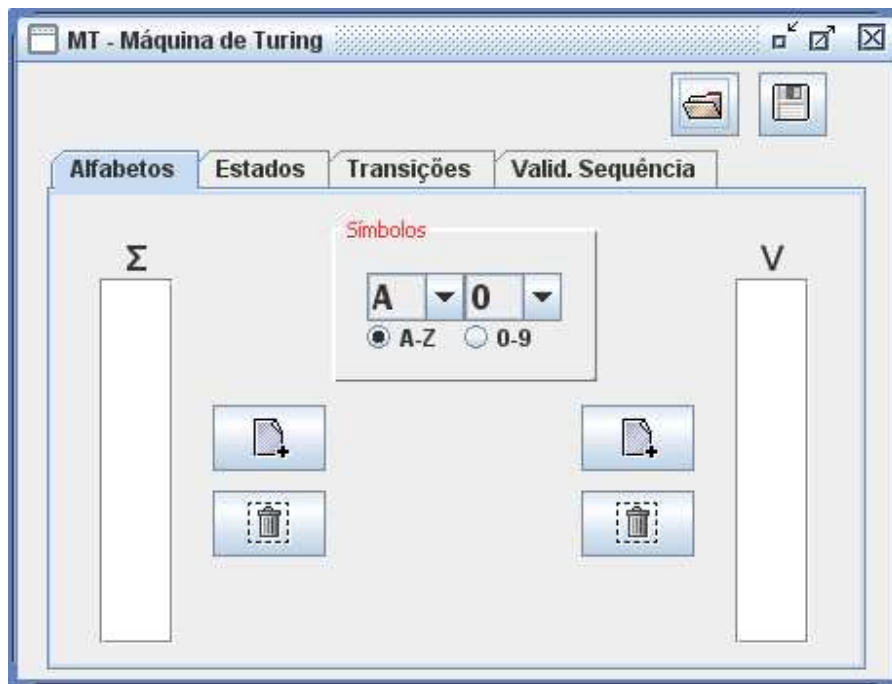


Figura 3. Janela para criação MT.

A ajuda atual (Figura 4) é composta apenas por um texto explicativo, onde todas as informações estão dispostas em um único documento. Neste documento existe uma explicação sobre os modelos formais e poucas informações com relação às funcionalidades do sistema. Caso o usuário tenha dificuldade de compreender o funcionamento de uma determinada funcionalidade o único modo de conseguir um suporte é através de um contato (via e-mail) direto com algum dos desenvolvedores através da opção sobre no menu ajuda. Em alguns casos o usuário pode simplesmente não conseguir utilizar a ferramenta por falta de uma documentação de ajuda adequada.

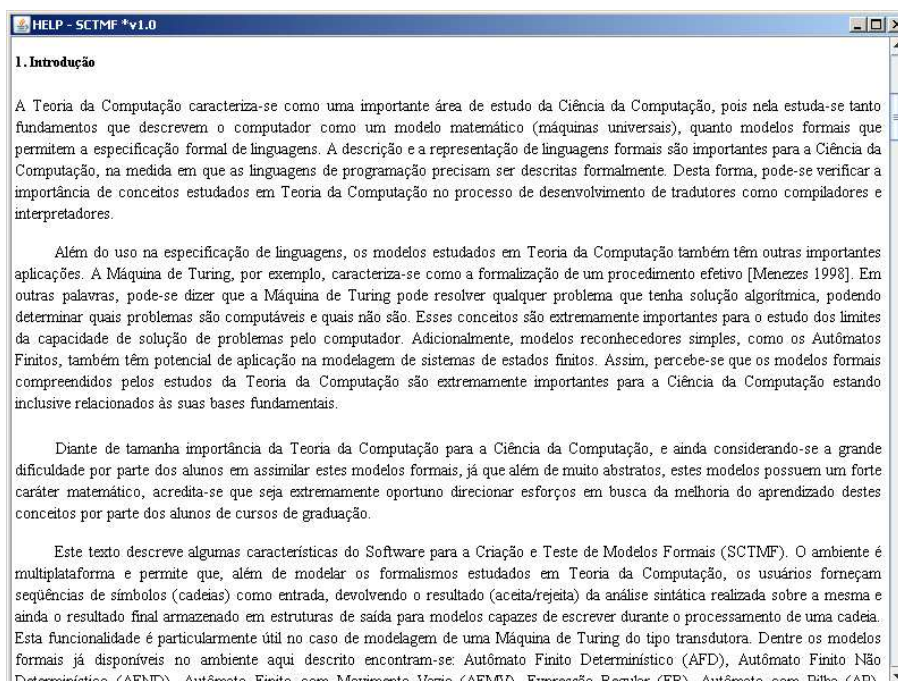


Figura 4. Ajuda SCTMF atual.

Dada a importância de um mecanismo de suporte que ofereça ao usuário com dificuldades, informações referentes à correta utilização do sistema e de suas funcionalidades, torna-se essencial que o ambiente possua um módulo que atenda a esta necessidade, fornecendo informações sobre os modelos aplicados e sua respectiva funcionalidade.

4. O módulo HELP!

No desenvolvimento foram utilizados os padrões de desenvolvimento para web, também conhecidos como *Web Standards*. Estes padrões (normas e diretrizes) foram criados pelo Consórcio *World Wide Web* (W3C) visando à padronização no desenvolvimento em ambiente web. Quando estas diretrizes são seguidas no desenvolvimento de uma página/site, pode-se dizer que esta aplicação é do tipo *Cross-browser*, ou seja, ela consegue ser acessada através de diversos navegadores sendo apresentada sempre da mesma maneira, garantido assim uma maior acessibilidade entre os browsers. Outra vantagem em utilizar estes padrões é o desempenho, por exemplo, o simples fato de organizar a aplicação em camadas (HTML, CSS, JavaScript) aumenta o desempenho no sentido de que os *browsers* guardam partes da aplicação em seu *cache* evitando ter que carregar novamente o mesmo arquivo.

Um exemplo prático da vantagem de desenvolver *web* utilizando *Web Standards* é o da reformulação do site da ESPN (Entertainment and Sport Programming Network) conforme [Budd et al. 2006] cada página teve uma redução média de cinquenta kilobytes por página, com isso a economia de banda da internet da empresa ficou em torno de dois terabytes por dia, ou seja, setecentos terabytes por ano.

Aplicações desenvolvidas para a internet geralmente são acessadas por pessoas com máquinas diferentes, com configurações diferentes, com sistemas diferentes, ou seja, é de extrema complexidade conseguir saber qual o usuário estará utilizando nossa aplicação. As Tabelas 1 e Tabela 2 apresentam um dos motivos desta complexidade, a crescente diversidade de navegadores. Com base nas informações destas tabelas é possível comprovar a importância de que uma aplicação desenvolvida para a *internet* possa ser acessada por diversos *browsers* em diferentes contextos. As informações destas tabelas foram extraídas de [W3Schools 2009], site oficial W3C com manuais de referência, tutoriais sobre as tecnologias recomendadas e referências sobre estatísticas da *web*.

Tabela 1. Estatística de utilização de navegadores

Data	IE6	IE7	IE8	Firefox	Chrome	Safari	Opera
Julho 2009	14,4%	15,9%	9,1%	47,9%	6,5%	3,3%	2,1%
Julho 2008	25,3%	26,4%	-	42,6%	-	2,5%	1,9%
Julho 2007	36,9%	20,1%	-	34,5%	-	1,5%	1,9%
Julho 2006	56,3%	1,9%	-	25,5%	-	-	1,4%

Tabela 2. Descrição dos navegadores

IE	Microsoft Internet Explorer
Firefox	Firefox (identificado como Mozilla antes de 2005)
Chrome	Google Chrome

Safari	Safari (e Konqueror. Ambos identificados como Mozilla antes de 2007)
Opera	Opera

A seguir serão apresentadas as tecnologias utilizadas e seus respectivos papéis no desenvolvimento deste trabalho:

- XHTML (eXtensible Hypertext Markup Language): Uma reformulação da linguagem de marcação HTML (Hypertext Markup Language) baseada em XML (eXtensible Markup Language), é das uma recomendações do W3C para publicação *web*. Foi utilizada no desenvolvimento do *Help!* na marcação e descrição de todo o conteúdo. A principal vantagem de se utilizar XHTML como linguagem base é o fato de ser facilmente interpretada por qualquer dispositivo, independente da plataforma utilizada, melhorando a acessibilidade (facilidade de acesso).
- CSS (Cascading Style Sheets): Conhecida também como folha de estilos em camadas, a CSS é uma linguagem utilizada para definir a apresentação de documentos editados em uma linguagem de marcação, como XML, HTML ou XHTML. Sua principal vantagem é a possibilidade de separação entre conteúdo e apresentação. A utilização desta tecnologia está presente na ajuda em todo o conteúdo referente à apresentação, facilitando a manutenção e possíveis trocas de padrões visuais. O *layout* (interface) do módulo *HELP!* foi completamente desenvolvido utilizando-se CSS juntamente com uma técnica chamada de Layout Elástico, esta técnica pode ser resumida em garantir que a interface do sistema seja bem apresentada independente da resolução que o usuário esteja utilizando. A Tabela 3 mostra as resoluções mais utilizadas atualmente por usuários da web segundo *W3Schools*.

Tabela 3. Resoluções de tela

Data	Superior	1024x768	800x600	640x480	Desconhecido
Janeiro 2009	57%	36%	4%	0%	3%

- JavaScript ou JS: Foi desenvolvida com o intuito de adicionar interatividade a documentos HTML, ela é uma linguagem de programação interpretada e não compilada. Quando utilizada em conjunto com CSS e HTML é conhecida como DHTML (Dynamic HTML), permitido que os conteúdos possam ser modificados dinamicamente. Durante o desenvolvimento do módulo de ajuda a linguagem JS foi utilizada para a criação do sistema de navegação, onde o usuário pode, de forma dinâmica, acessar todo conteúdo disponível e em algumas funções repetitivas como, por exemplo, as abreviações geradas automaticamente.

No módulo ajuda os modelos formais foram organizados e classificados de acordo com sua estrutura hierárquica (Figura 5) na Hierarquia de Chomsky. Para cada modelo foi desenvolvida uma página contendo: uma introdução inicial sobre o modelo em questão, uma definição formal matemática do modelo, um exemplo incluindo ilustração através de grafos, e um exemplo prático e detalhado explicando passo a passo como criar e avaliar um modelo formal utilizando a ferramenta SCTMF.

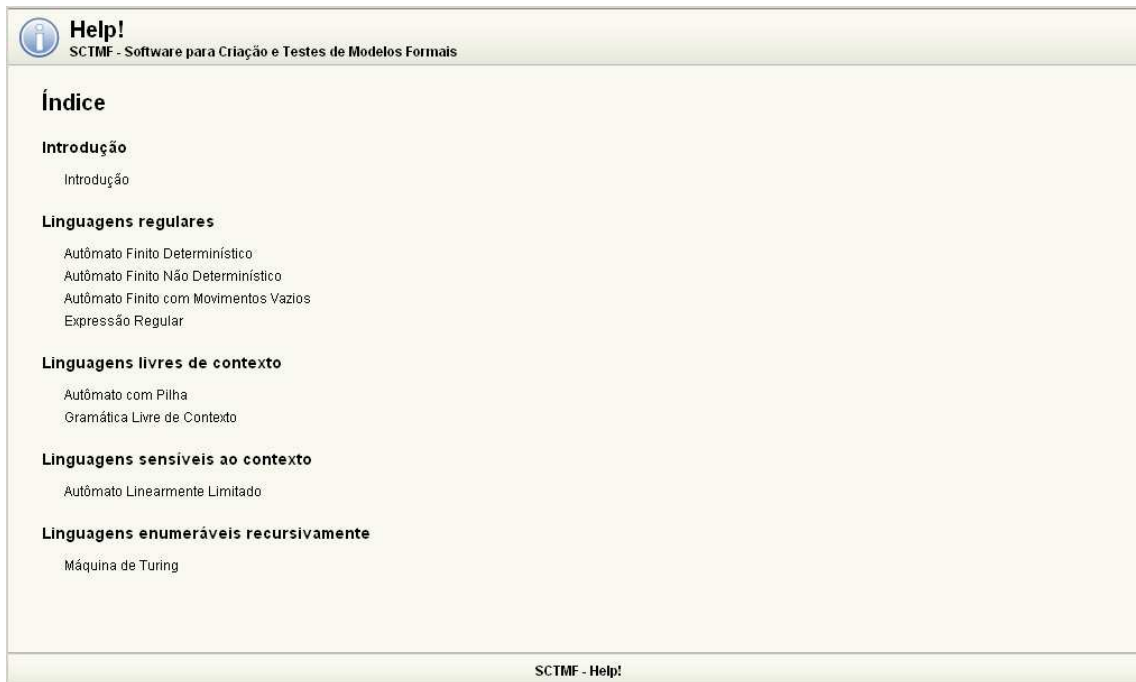


Figura 5. Página inicial módulo ajuda.

Com o intuito de facilitar a navegação e garantir uma maior usabilidade (facilidade com que determinada funcionalidade pode ser utilizada), todas as páginas possuem um sistema de índice, contendo atalhos para os conteúdos da página, agilizando a procura de alguma informação específica. Outro detalhe importante para a navegação é a barra inferior, esta trabalha no sentido de uma navegação global, onde a qualquer momento o usuário pode retornar ao índice principal ou alternar para o modelo anterior ou próximo.

4.1. Conceitos de usabilidade aplicados

Durante o processo de desenvolvimento alguns conceitos de usabilidade foram sendo incorporados na aplicação. Por exemplo, o sistema de navegação foi projetado para que com no máximo dois cliques o usuário possa encontrar a funcionalidade desejada, atendendo assim, uma das regras básicas de usabilidade quanto à facilidade de navegação onde um usuário deve encontrar a informação desejada em no máximo três cliques [Nielsen 2000].

Outra regra básica que foi atendida foi a de clareza na arquitetura da informação, pois no início de cada página foram adicionados índices que funcionam como “mini mapas do site” onde o usuário facilmente pode se localizar dentro da seção. As cores utilizadas foram cores frias e leves o que atende a mais uma regra, a simplicidade, fazendo com que o usuário canse-se menos ao utilizar o módulo tornando a experiência com a ferramenta mais prazerosa. Segundo mais uma das regras (a relevância do conteúdo) foram adicionados botões de impressão em todas as seções, facilitando caso o usuário deseje imprimir determinado modelo. No caso da impressão alguns elementos como links e botões são omitidos, sendo impresso somente o conteúdo.

Todas as seções da ajuda seguem uma mesma linha de *design*, ou seja, existe um padrão para cores e tamanhos de letras, posicionamento dos botões e atalhos. Enfim esta padronização implica em diminuir a curva de aprendizado do usuário, pois quando ele conhece a página (*layout*) de um determinado modelo ao conhecer a página de outro modelo ele terá os mesmos elementos

presentes na página anterior, acelerando o tempo de assimilação, pois ele não tem a necessidade de aprender tudo novamente.

5. Considerações finais

O SCTMF é uma ferramenta de grande importância no auxílio do aprendizado de Teoria da Computação em especial modelos formais, pois contempla todas as classes da Hierarquia de Chomsky. Com a sua utilização alunos e professores podem de uma forma fácil e ágil criar e testar modelos formais tornando a experiência de aprendizado algo mais prático e menos teórico, logo, esta facilidade contribuirá para um aumento no interesse pelo assunto. Outra característica importante da ferramenta é sua independência de plataforma, o que aumenta sua abrangência de usuários.

Dada a importância do projeto SCTMF e a complexidade dos modelos por ele suportados, um mecanismo de ajuda torna-se uma funcionalidade imprescindível. O Help! desenvolvido neste trabalho vem como solução para auxiliar usuários com dificuldades na utilização da ferramenta. Apesar de não ser utilizada com a mesma tecnologia do SCTMF (Java), o módulo ajuda, também pode ser considerado multiplataforma sendo dependente apenas de um browser encontrado na maioria dos sistemas operacionais atualmente utilizados.

5.1. Trabalhos futuros

Pretende-se adicionar ao ambiente SCTMF um módulo que permita ao usuário criar todos os tipos de modelos reconhecidos disponíveis na ferramenta visualmente, através de construção de diagramas. Este novo módulo visa aumentar ainda mais a interação do usuário com o ambiente, tornando a experiência do usuário mais intuitiva. Logo, grande parte do mecanismo de ajuda necessitará de atualizações referentes a esta nova funcionalidade.

Agradecimentos

A minha esposa Talita por me apoiar e incentivar nos momentos de estudos, a empresa Belatriz por acreditar em meu potencial, ao orientador Professor Mestre Yandre e a equipe de desenvolvimento Bruno, Fabio, Michel e Rafael pela oportunidade de contribuir com o projeto SCTMF.

Referências

- Bortolotto, F. S. e Costa, Y. M. G. (2009). “Implementação de módulo para conversão entre modelos formais no sistema para a criação e testes de modelos formais”, em XVIII Encontro Anual de Iniciação Científica – PIBIC-CNPq.
- Budd, A.; Collision, S.; Moll, C. Criando páginas Web com CSS – Soluções avançadas para padrões Web. São Paulo: Pearson Prentice Hall, 2006.
- Costa, Y. M. G., Meneses, R. C. e Uber, F. R. (2008). “Uma ferramenta para auxílio didático no ensino de teoria da computação”, em Workshop sobre educação em computação - XXVIII Congresso da Sociedade Brasileira de Computação.

- Meneguello, B. K., Bortolotto, F. S., Miranda, M. M. B. e Costa, Y. M. G. (2009). “Desenvolvimento de um editor gráfico interativo para o sistema para a criação e teste de modelos formais (SCTMF)”, em XVIII Encontro Anual de Iniciação Científica – PIBIC-CNPq.
- Meneses, R. C. e Costa, Y. M. G. (2007). “Desenvolvimento de uma ferramenta multiplataforma para criação e teste de modelos formais”, em XVI Encontro Anual de Iniciação Científica – PIBIC-CNPq.
- Menezes, P. B. (1998) “Linguagens Formais e Autômatos”, Sagra Luzzatto, 2a. Edição.
- Miranda, M. M. B. e Costa, Y. M. G. (2009). “Inclusão do Autômato Linearmente Limitado no software para criação e teste de modelos formais”, em XVIII Encontro Anual de Iniciação Científica – PIBIC-CNPq.
- Nielsen, J. (2000) *Projetando Websites*, trad. Ana Gibson. Rio de Janeiro: Campus.
- “W3Schools”, <http://www.w3schools.com>, Acessado em Agosto de 2009.