

Ferramenta Didática Interativa Utilizando a Linguagem NCL: Algoritmo de Quine-McCluskey

H. V. Mattos¹, T. M. de Souza¹, A. S. de la Vega¹, D. C. M. Saade²

¹Grupo PET-Tele – pet.telecom.uff.br

²Laboratório MídiaCom – www.midiacom.uff.br

Universidade Federal Fluminense – Escola de Engenharia

Departamento de Engenharia de Telecomunicações

Rua Passo da Pátria, 156 / Bl. D / Sala 504 – Niterói – RJ

Resumo

Este artigo aborda a criação de um aplicativo interativo para o sistema de TV Digital brasileiro, com o intuito de ser utilizado como ferramenta didática para o aprendizado do *Algoritmo de Quine-McCluskey*. O *Algoritmo de Quine-McCluskey*, é um método de minimização de funções lógicas [1]. Funções lógicas são utilizadas na construção de circuitos digitais, os quais são amplamente empregados no ramo de Telecomunicações. O uso da linguagem NCL para o desenvolvimento do aplicativo justifica-se pelo fato da linguagem ser capaz de gerenciar eficientemente várias mídias diferentes e, portanto, mostrar-se adequada para o desenvolvimento de aplicativos didáticos interativos.

Palavras-chaves: Ferramenta didática interativa, Linguagem NCL, TV Digital, Minimização de funções lógicas, Algoritmo de Quine-McCluskey.

1 Introdução

Diante da determinação do governo brasileiro sobre o padrão de TV Digital adotado no país, viu-se uma larga oportunidade para a resolução do problema da inclusão digital da população. O padrão adotado no Brasil, é baseado no padrão japonês ISDB-T, com algumas contribuições de pesquisadores brasileiros. Na área de *software*, pode ser citada a inclusão de um *middleware* formado por uma combinação das linguagens NCL [2], Java [3] e Lua [4]. A linguagem NCL tem a característica de gerenciar eficientemente mídias de diversos tipos, permitindo um alto grau de interatividade. Com isso, espera-se que milhões de lares brasileiros consigam se comunicar de forma digital, ampliando a inclusão digital no país.

Aproveitando a oportunidade dada pela adoção em massa da TV Digital, espera-se que seja incentivado o seu uso e a difusão de conhecimentos.

Nesse contexto, espera-se ainda que o aprendizado interativo possa ser amplamente difundido.

O projeto realizado teve como objetivo o desenvolvimento de uma ferramenta didática interativa, versando sobre o *Algoritmo de Quine-McCluskey*, baseada em linguagem NCL e distribuída de forma gratuita. Através de um exemplo de minimização, o aplicativo procura ilustrar o emprego do método e sua importância.

Esse trabalho é organizado da seguinte forma. A Seção 2 apresenta um breve resumo sobre o algoritmo, enquanto a justificativa para o uso das linguagens NCL e Lua encontra-se na Seção 3. Uma breve descrição do modelo da linguagem NCL é descrito na Seção 4. O aplicativo desenvolvido é descrito na Seção 5. As conclusões e as sugestões para trabalhos futuros encontram-se na Seção 6.

2 O algoritmo implementado

No ramo de Telecomunicações, circuitos digitais são largamente utilizados, nas mais diversas aplicações.

Os circuitos digitais podem ser divididos em dois grandes grupos: circuitos combinacionais e circuitos sequenciais.

Circuitos digitais combinacionais podem ser interpretados como implementações de funções lógicas booleanas. Uma mesma função pode ser descrita por diversas expressões lógicas. Quanto menor for a expressão lógica associada à função, menor será o circuito digital implementado. Assim, deve ser feito um esforço para minimizar as expressões lógicas envolvidas.

Dada a tabela verdade associada a uma função lógica, pode-se escrever uma expressão que sintetize seus valores lógicos “1” ou “0”. Uma combinação que empregue a operação lógica AND de todas as variáveis envolvidas na função, e que sintetize um de seus valores lógicos “1”, é denominado mintermo. Mintermos podem ser combinados logicamente, eliminando variáveis dessa combinação e, portanto, minimizando a expressão final. Tais combinações e os próprios mintermos, que representam um ou mais valores lógicos “1”

simultaneamente, recebem a denominação de implicantes. Um implicante que não pode ser mais simplificado recebe a denominação de implicante primo. Desse modo, minimizar uma expressão lógica é equivalente a encontrar um conjunto ótimo de implicantes primos que realize a função desejada com o menor custo.

Na busca do conjunto ótimo de implicantes primos, um método gráfico largamente utilizado é o Mapa de Karnaugh [1]. Uma vantagem desse método é a sua simplicidade de execução. Porém, ele apresenta duas grandes desvantagens. Primeiramente, sua complexidade aumenta com o aumento do número de variáveis. Para funções contendo seis ou mais variáveis o método começa a se tornar impraticável. Além disso, o método envolve um alto grau de subjetividade na sua execução.

Uma alternativa algorítmica é um método tabular denominado de Método dos Implicantes Primos ou Método de Quine-McCluskey. Em uma primeira fase, os mintermos são agrupados em implicantes maximamente simplificados, denominados implicantes primos. Na fase seguinte, é selecionado um subconjunto de implicantes primos que seja suficiente para representar todos os mintermos da função [1].

Levando em conta o número de expressões mínimas equivalentes, podem ser identificados dois grupos de expressões lógicas. Para o primeiro delas, as quais possuem uma expressão mínima única, o algoritmo básico de Quine-McCluskey é suficiente. Para o segundo grupo, onde as expressões lógicas possuem mais de uma expressão mínima, é necessário aplicar um método de escolha adicional. O método de Petrick [1] é comumente aplicado nesses casos.

O aplicativo desenvolvido aborda apenas o algoritmo básico de Quine-McCluskey.

3 Escolha das linguagens

Diversas implementações do algoritmo de Quine-McCluskey podem ser encontradas, usando linguagens procedurais (C++ [5], Java [6], Matlab [7], Pascal [8], Perl [9], Python [10]). Tais implementações permitem uso local ou via internet. Nesses casos, diversas interfaces gráficas podem ser utilizadas para trocar dados com o usuário. Porém, a integração de NCL com Lua viabiliza uma melhoria na interface com o usuário. Enquanto Lua realiza a parte de implementação procedural do cálculo do algoritmo, NCL é capaz de gerenciar eficientemente várias mídias diferentes para interagir com o usuário. Assim sendo, espera-se que o uso conjunto das linguagens NCL e Lua seja bastante adequado para o desenvolvimento de aplicativos didáticos interativos.

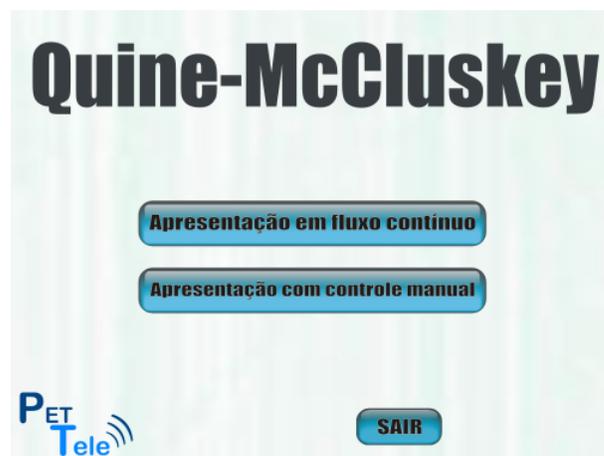


Figura 1: Quadro inicial do aplicativo.

4 A linguagem NCL

NCL é uma linguagem fundamentada no paradigma declarativo de programação.

A linguagem NCL, inserida no modelo NCM (*Nested Context Model*), possui os conceitos de nós e elos e, através destes conceitos, podem-se descrever documentos hipermídia, que são documentos onde são reunidos diversos tipos de mídia. Tais mídias podem interagir através de ações por parte do usuário.

Na estrutura do modelo NCM, cada nó pode ser também um conjunto de nós e elos e, quando isso ocorre, denomina-se nó de composição ou nó de contexto. Tal construção torna o documento mais organizado e seu código menor e mais eficiente, principalmente para documentos extensos. Portanto, ao descrever um documento NCL, estamos lidando, basicamente, com dois tipos distintos de nós: i) nós de mídia, que descrevem imagem, texto, áudio, vídeo e demais objetos de mídia e ii) nós de contexto, que descrevem uma coleção de nós de mídia e seus elos associados.

5 O aplicativo

O *middleware* do Sistema Brasileiro de TV Digital (SBTVD) utiliza linguagens de programação com dois tipos de paradigma. As linguagens Java e Lua possuem paradigma procedural, enquanto a linguagem NCL possui paradigma declarativo. No desenvolvimento do aplicativo, foi utilizada a linguagem NCL.

A interface com o usuário foi desenvolvida para ser a mais confortável possível, pois foi assumido que o usuário deverá portar apenas um controle remoto para um televisor digital. A Figura 1 ilustra o quadro inicial do aplicativo.

O aplicativo implementa dois tipos distintos de navegação: “Apresentação em fluxo contínuo” e “Apresentação com controle manual”. Na apresentação em fluxo contínuo, o usuário não necessita, obrigatoriamente, pressionar

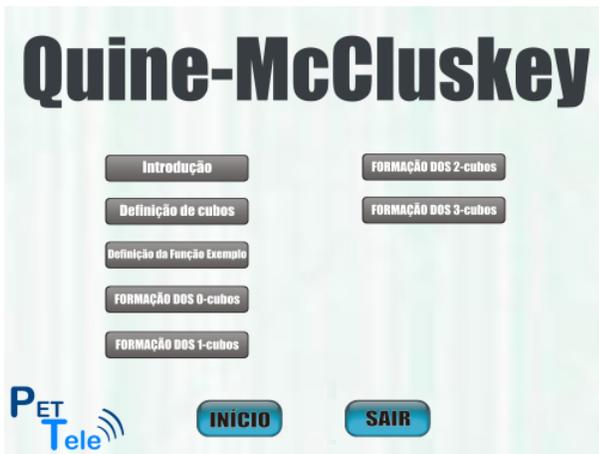


Figura 2: Quadro “Apresentação com controle manual”.

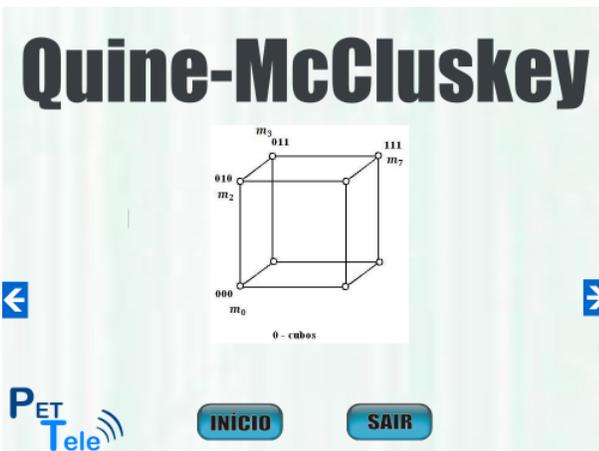


Figura 3: Quadro intermediário da ferramenta.

o botão “avançar” para que a aplicação se desenvolva. A aplicação irá avançar automaticamente a cada tela, assim que a tela corrente finalizar o seu áudio. Na apresentação com controle manual, o usuário pode escolher o ponto de entrada no aplicativo e é obrigatório que o usuário pressione o botão “avançar” para que a aplicação possa alternar para a próxima tela. A justificativa para a existência de um modo manual é que o usuário tenha mais tempo para compreender todo o conteúdo apresentado em cada tela. A Figura 2 apresenta o quadro que aparece após a escolha de “Apresentação com controle manual”.

Os botões “retroceder”, “avançar”, “início” e “sair”, podem ser acessados a qualquer instante, em qualquer dos dois modos, e auxiliam o usuário para uma melhor navegação. A Figura 3 apresenta um quadro onde podem ser visualizados todos esses botões.

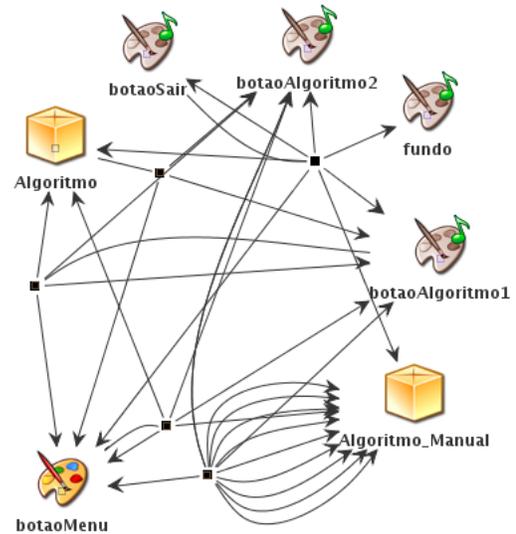


Figura 4: Visão estrutural do aplicativo.

Apresentado através de 63 telas, com uso de figuras, texto e áudio, o aplicativo mostra ao usuário, em passos detalhados, como iniciar, desenvolver e finalizar o uso do método de Quine-McCluskey, utilizando um exemplo fixo de minimização.

Na implementação do aplicativo em linguagem NCL, a “Apresentação em fluxo contínuo” e a “Apresentação com controle manual” foram separadas em dois contextos distintos. Cada contexto possui seus próprios objetos de mídia e suas próprias regras de sincronismo e apresentação.

Durante a implementação foi escolhido que o áudio iria fazer o sincronismo com todos os outros objetos de mídia existentes na apresentação. Desse modo, o áudio foi subdividido em seções e estas comandam a entrada e saída de figuras e textos na tela. A Figura 4, elaborada no Editor de Documentos NCL (COMPOSER) [11], mostra a visão estrutural do aplicativo.

Os botões “início” (botaoMenu) e “sair” (botaoSair) foram implementados fora dos contextos. Isso foi feito para que se pudesse utilizar o recurso “re-uso de nós”, técnica disponibilizada pela linguagem NCL, que facilita a programação e aumenta a eficiência do aplicativo.

Dado um aplicativo didático, espera-se que o mesmo apresente, pelo menos, duas características importantes. Primeiramente, ele deve se basear em uma operação interativa com o usuário. Além disso, deve ser permitido ao usuário experimentar várias situações diferentes. Conforme descrito acima, o aplicativo desenvolvido possui um bom grau de interatividade com o usuário, permitindo que o mesmo navegue facilmente entre as diversas etapas do algoritmo abordado. Quanto à variabilidade de situações, pretende-se realizá-la em versões futuras do aplicativo, conforme será citado a seguir.

6 Conclusões

Espera-se que o advento da TV Digital no Brasil gere muitas oportunidades para todos os segmentos envolvidos, principalmente para a população, que, diante do cenário de baixa inclusão digital que reina no país, terá oportunidade de se interar das principais inovações educacionais e tecnológicas.

Sob esse aspecto, este projeto torna-se importante no quesito educação, por se tratar de um *software* gratuito e que pode ser utilizado em qualquer aparelho de TV digital. Os estudantes terão a oportunidade de adquirir este conhecimento em casa, diante de um televisor e com apenas o controle remoto em mãos.

Uma vez que a versão atual do aplicativo caracteriza-se apenas como um manual *online* e interativo, o seu código fonte foi baseado apenas sob o paradigma declarativo de programação, utilizando a linguagem NCL. Apesar da interface com o usuário ter sido desenvolvida com o intuito de ser o mais confortável possível, até a elaboração desse artigo não foi possível avaliá-la adequadamente com os alunos do curso de graduação em Engenharia de Telecomunicações.

O próximo passo para uma evolução do aplicativo será permitir que o usuário apresente diferentes expressões lógicas a serem minimizadas. Para isso, será necessário implementar o algoritmo de Quine-McCluskey. Porém, a linguagem NCL, possuindo um paradigma declarativo, não se presta a esse tipo de implementação. Uma vez que o *middleware* da TV digital possui uma linguagem com paradigma procedural (Lua) e uma API de ligação (NCLua), pretende-se implementar o citado algoritmo na linguagem Lua. Assim sendo, o usuário poderá estudar toda a teoria através do manual interativo, exercitar a minimização de expressões manualmente e, por fim, verificar o resultado utilizando a parte procedural do aplicativo.

Agradecimentos

O grupo PET-Tele faz parte do Programa de Educação Tutorial (PET), financiado pelo Ministério da Educação (MEC).

Os autores agradecem ao aluno Felipe Sanches da Costa Maia Alves, integrante do grupo PET-Tele, pela elaboração dos elementos gráficos do aplicativo.

Referências

- [1] F. J. Hill and G. R. Peterson. *Introduction to Switching Theory and Logical Design*. John Wiley, New York, NY, 3rd edition, 1981.
- [2] L. F. G. Soares and S. D. J. Barbosa. *Programando em NCL 3.0: Desenvolvimento de aplicações para o middleware Ginga*. Campus, Rio de Janeiro, RJ, 2009.
- [3] H. M. Deitel and P. J. Deitel. *Java: How to Program*. Prentice-Hall, 7th edition, 2007.
- [4] R. Ierusalimschy. *Programming in Lua*. Lightning Source Inc, Rio de Janeiro, RJ, 2nd edition, 2006.
- [5] <http://sourceforge.net/projects/qmcs/>.
- [6] <http://www.mathcs.bethel.edu/~gossett/DiscreteMathWithProof/QuineMcCluskey.html>.
- [7] <http://automatics.hit.bg/#minBool>.
- [8] <http://sourceforge.net/projects/quinessence/>.
- [9] <http://search.cpan.org/~kulp/Algorithm-QuineMcCluskey-0.01/lib/Algorithm/QuineMcCluskey.pm>.
- [10] <http://pypi.python.org/pypi/qm/0.2>.
- [11] <http://www.gingancl.org.br>.