Desenvolvimento de um software para solucionar mapas de Karnaugh pelo método de Quine-Mccluskey

Tiago Leite Brito

Curso de Engenharia de Computação – Universidade Federal do Pampa (UNIPAMPA)
Avenida Maria Anunciação Gomes, nº1650 – Bagé – RS – Brasil

atoumdesign@gmail.com

1. Introdução

Através da necessidade de aprender como é utilizado o método de Quine-Mccluskey para a resolução de mapas de Karnaugh em sala de aula, foi desenvolvido uma solução prévia que auxiliava em visualizar as comparações realizadas. Esta solução atendia razoavelmente bem para a resolução dos exercícios mas não trazia detalhes importantes como a tabela de cobertura por exemplo.

Como trabalho de conclusão de semestre foi cogitado ampliar as funcionalidades deste software, podendo determinar quantas variáveis serão utilizadas e torná-lo uma ferramenta didática para o auxilio em aulas de técnicas digitais.

2. Metodologia

Para o desenvolvimento do software foi utilizado o framework BscScrum, desenvolvido na disciplina de Engenharia de software e ainda em desenvolvimento, para poder medir o desempenho do desenvolvimento e a eficácia do resultado, por se tratar de um período de 24 dias da concepção a entrega.

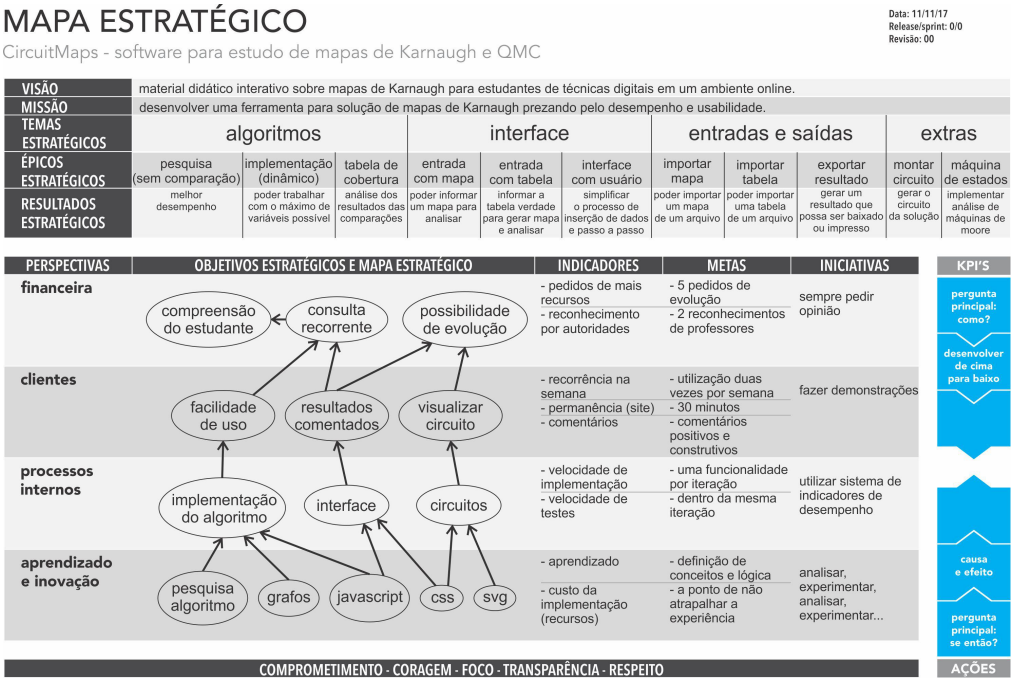

Figura 1. mapa estratégico do projeto

Conforme descrito na figura 1, foram definidos os temas principais e uma breve descrição do que é esperado em seus épicos subsequentes. Também foi estabelecidos objetivos a serem alcançados e metas a serem atingidas em um âmbito geral no desenvolvimento. A sequência das atividades de produção do software foi descrito em seu product backlog, podendo ser visto na figura 2.

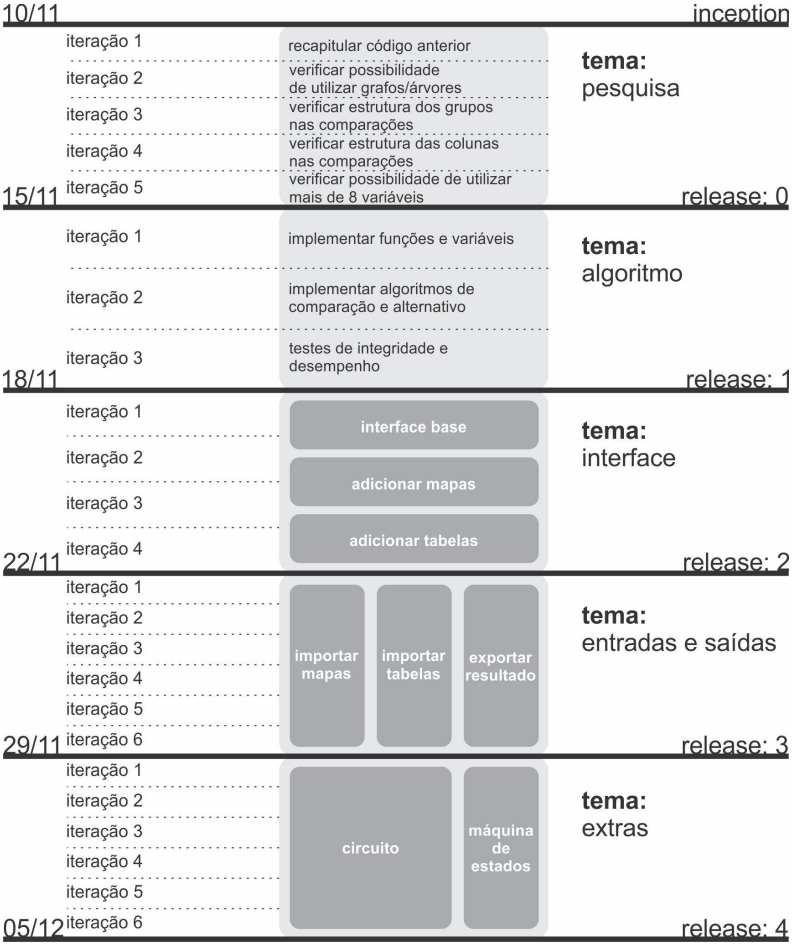

Figura 2. product backlog inicial do projeto

Com um proposta de fácil acesso, a ferramenta foi desenvolvida em javascript e com foco em ser hospedada em algum provedor gratuito para as primeiras avaliações.

3. Implementação

Inicialmente foi reavaliado o código utilizado anteriormente observando os pontos que deveriam ser descartados, mantidos, evoluídos e o que precisava ser desenvolvido. Neste período de elaboração foi definido utilizar funções dinâmicas para poder testar o máximo de variáveis possíveis, foi rejeitada a utilização da busca em grafos por causa do curto tempo de desenvolvimento, mas foi tentado trazer uma possível alternativa de melhoria no algoritmo de comparações do método Quine-Mccluskey.

Foram cogitadas quatro hipóteses: a primeira tentando eliminar as comparações desnecessárias ao máximo, a segunda tentando começar as comparações do fim para o inicio, a terceira utilizando uma análise em profundidade de um toróide em 4D contendo o mapa de Karnaugh em seu interior e uma busca direta em um banco de dados das

respostas. Novamente por questões de tempo foi definida a primeira por ser mais facilmente aplicada.

A implementação do método de forma tradicional foi implementada da seguinte forma:

1. É gerado um vetor denominado mapa com todos os valores do mapa ou da tabela verdade em suas respectivas posições.

2. Depois de uma análise prévia, é preenchido um vetor denominado resultado com as posições e binários, assim como o estabelecimento dos grupos e colunas que serão utilizados conforme o número de variáveis (ver figura 3).

3. É realizada comparações entre os grupos de uma coluna e gravando na próxima coluna dentro do mesmo vetor resultado.

4. As informações do vetor resultado, depois de todas as comparações realizadas, abastecem o vetor cobertura (ver figura 4).

5. É realizado uma analise neste vetor cobertura que contém somente os implicantes primos da seguinte forma, é contado o número de uns em cada posição e a que tiver somente a contagem um já é automaticamente seleciona como implicante primo essencial e eliminado deste vetor.

6. O processo anterior é realizados recursivamente até não restarem mais somas com resultado um, ou caso restem algum implicante primo estes são avaliados com os que tem o mais número de uns contidos.

7. A solução do problema é mostrada como a lista de implicantes primos essenciais.

Figura 3. vetor de objetos resultado

```
» cobertura2
← ▾ […]
    ▸ 0: Object { binario: "_0000", posicao: "0 16", validos: […] }
    ▸ 1: Object { binario: "0_0_0", posicao: "0 8 2 10", validos: […] }
    ▸ 2: Object { binario: "0__10", posicao: "2 10 6 14", validos: […] }
    ▸ 3: Object { binario: "01__0", posicao: "8 12 10 14", validos: […] }
    ▸ 4: Object { binario: "10_0_", posicao: "16 20 17 21", validos: […] }
    ▸ 5: Object { binario: "_01_1", posicao: "5 21 7 23", validos: […] }
    ▸ 6: Object { binario: "__101", posicao: "5 21 13 29", validos: […] }
    ▸ 7: Object { binario: "_011_", posicao: "6 22 7 23", validos: […] }
    ▸ 8: Object { binario: "__110", posicao: "6 22 14 30", validos: […] }
    ▸ 9: Object { binario: "_110_", posicao: "12 28 13 29", validos: […] }
    ▸ 10: Object { binario: "_11_0", posicao: "12 28 14 30", validos: […] }
    ▸ 11: Object { binario: "1___1", posicao: "17 25 21 29 19 27 23 31", validos: […] }
    ▸ 12: Object { binario: "1_1__", posicao: "20 28 22 30 21 29 23 31", validos: […] }
      length: 13
    ▸ __proto__: Array []
```

Figura 4. vetor de objetos cobertura

A realização do algoritmo alternativo se baseia em determinar quando as comparações que não trarão resultados ocorrerão e evitá-las. Foi desenvolvido um algoritmo inicial para validar a ideia, onde era realizado o cálculo das posições dinamicamente de acordo com o número de variáveis demonstrados na figura 5.

```
var g0 = 0;
var g1 = [1,2,4,8,16,32,64,128];
var g2 =
[3,5,6,9,10,12,17,18,20,24,33,34,36,40,48,65,66,68,72,80,
96,129,130,132,136,144,160,192];
var g3 =
[7,11,13,14,19,21,22,25,26,28,35,37,38,41,42,44,49,50,
52,56,67,69,70,73,74,76,81,82,84,88,97,98,100,104,112,131,1
33,134,137,138,140,145,146,148,152,161,162,168,176,193,19
4,196,200,208,212,224];
var g4 =
[15,23,27,29,30,39,43,45,46,51,53,54,57,58,60,71,75,77,
78,83,85,86,89,90,92,99,101,102,105,106,108,113,114,116,12
0,135,139,141,142,147,149,150,153,154,156,163,165,166,169
,170,172,177,178,180,184,195,197,198,201,202,204,208,210,
212,216,225,226,228,232,240];


function coluna(v) {
    //grupo0                                    grupo1 = 1
    if (mapa[0] == 1 || mapa[0] == "x") {
        var tex = document.querySelector("#c0g0");
        tex.innerHTML = "(0)";
    }

    //grupo1
    var texto = "";                             grupo1 = n
    for (let i = 0; i < v; i++) {
        var temp = parseInt(g1[i]);
        if ((mapa[temp] == 1) || (mapa[temp] == "x")) {
            texto = texto+"("+temp+")<br>";
        }
        var tex = document.querySelector("#c0g1");
        tex.innerHTML = texto;
    }

    //grupo2                                    grupo2 = (n-1)n / 2
    var v2 = ((v-1)*v)/2;
    texto = "";
    for (let i = 0; i < v2; i++) {
        var temp = parseInt(g2[i]);
        if ((mapa[temp] == 1) || (mapa[temp] == "x")) {
            texto = texto+"("+temp+")<br>";
        }
        var tex = document.querySelector("#c0g2");
        tex.innerHTML = texto;
    }

    //grupo3                                    grupo3 = Σ_{n=1}^{n} (n-2)(n-1) / 2
    var v3 = 0;
    for (let i = 0; i < v; i++) {
        v3 = v3+(((v-1)*v)/2);
    }
    texto = "";
    for (let i = 0; i < v3; i++) {
        var temp = parseInt(g3[i]);
        if ((mapa[temp] == 1) || (mapa[temp] == "x")) {
            texto = texto+"("+temp+")<br>";
        }
        var tex = document.querySelector("#c0g3");
        tex.innerHTML = texto;
    }
}
```

Figura 5. algoritmo validador da hipótese

Com a validação da hipótese, foi encontrado o primeiro obstáculo, conforme o número de variáveis aumentava os cálculo passam de divisões, para somas simples, depois somatório mais complexos e recaindo em integrais. Pelo pouco tempo restante para completar a tarefa, os cálculos foram substituídos pelos seus resultados em vetores, conforme a figura 6.

```
var coluna0 = [ ["0"],
                ["1","10","100","1000","10000"],
                ["11","101","110","1001","1010","1100","10001","10010","10100","11000"],
                ["111","1011","1101","1110","10011","10101","10110","11001","11010","11100"],
                ["1111","10111","11011","11101","11110"],
                ["11111"]];
var coluna1 = [ ["0_","_0","_00","_000","_0000"],
                ["_1","1_","_01","_10","10_","1_0"],
                ["_11","1_1","11_"],
                [],
                []];
var coluna2 = [ ["0__","_0_","__0"],
                ["__1","_1_","1__"],
                [],
                []];
var coluna3 = [ [],
                [],
                []];
var coluna4 = [ [],
                []];
```

```
var col0 = [[1,2,1],
            [1,3,3,1],
            [1,4,6,4,1],
            [1,5,10,10,5,1]];

var col1 = [[2,2],
            [3,6,3],
            [4,12,12,4],
            [5,20,30,20,5]];

var col2 = [[],
            [3,3],
            [6,12,6],
            [10,30,30,10]];

var col3 = [[],
            [],
            [4,4],
            [10,20,10]];

var col4 = [[],
            [],
            [],
            [5,5]];
```

Figura 6. algoritmo alternativo (WIP)

Com o prazo de entrega do software chegando ao seu final, foi decidido abortar esta funcionalidade, deixando para avaliação no próximo sprint planning e dar ênfase ao MVP (mínimo produto viável).

4. Conclusão

A aposta em um modelo alternativo de algoritmo de comparação se mostrou viável e com uma maior eficiência em testes preliminares, mas por falta de conhecimento prévio de algumas técnicas, acabou afetando o cronograma e subsequentemente a implementação de algumas funcionalidades do software.

As propostas de evoluções são inúmeras e realizáveis, como o término deste algoritmo alternativo, a implementação do algoritmo de comparação reversa quando a quantidade de uns no mapa é muito grande, a implementação do algoritmo de busca no banco de dados para agilizar o tempo de resposta para um número muito grande de variáveis, mas principalmente neste caso para servir de confirmação para os outros algoritmos. A implementação do algoritmo do toróide em 4D demanda uma pesquisa mais aprofundada em alguns conceitos matemáticos para poder validá-la o que a deixa em uma das últimas a serem exploradas.

A implementação do passo a passo com as explicações, a construção do circuito e exportação para o logisim também são facilmente implementáveis. A adição da análise de máquinas de estado e de outros componentes como lathes e flip-flops também já possuem esboços, mas requerem mais tempo de planejamento e implementação.