# Operadores Booleanos para a Classe de *NM*-conjuntos


Luiz Gonzaga da Silveira Júnior[1]
Wu, Shin-Ting[2]

Grupo de Computação de Imagens (GCI)
Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica (FEE)
Universidade Estadual de Campinas (UNICAMP)
C.P. 6101, Unicamp
13081-970 Campinas - SP

[1]e-mail : `gonzaga@dca.fee.unicamp.br`

[2]e-mail : `ting@dca.fee.unicamp.br`



**Abstract.** In this paper we present an algorithm for boolean set operations to NM-sets that fulfil three requirements: closeness, interactivity and efficiency.

We begin by showing that NM-sets are an extension of r-sets and are representable in TDM data structure. Then, a modified set of boolean operators under which the NM-sets are closed is presented. Finally, an implementation of these operators to an interactive process is discussed.


## 1 Introdução

A operação booleana é uma importante técnica de modelagem geométrica utilizada para a construção de objetos complexos a partir de operações de união, interseção e diferença (denotadas por $\cup$, $\cap$, e $-$, respectivamente) em objetos simples e de fácil especificação denominados primitivas.

A concepção de algoritmos de operação booleana está intimamente ligada ao esquema de representação e ao *espaço de modelagem* [7]. Sistemas de modelagem de sólidos baseados na representação CSG (*Constructive Solid Geometry*) combina primitivas sólidas simples tais como cones, cilindros, esferas e cubos para produzir objetos sólidos complexos. O algoritmo de combinação consiste em construir uma estrutura de árvore na qual as folhas representam as primitivas e os nós internos as operações booleanas regularizadas (denotadas por $\cup^*$, $\cap^*$ e $-^*$). Em esquemas de representações b-rep (*boundary representation*) os algoritmos de operações booleanas requerem a computação explícita das interseções entre dois modelos b-rep para criar um terceiro modelo, também b-rep.

A propriedade de fechamento é desejada para os operadores booleanos, pois garante a exclusão de objetos inválidos. Quando se trata de sólidos fisicamente realizáveis, equivalentes à variedades bidimensionais fechadas, Mäntylä [5] mostrou que, implementando os operadores booleanos com os operadores de Euler propostos por ele, os resultados são sempre representáveis pela estrutura *half-edge* e portanto, são variedades bidimensionais fechadas. Quando se trata de r-conjuntos, Requicha [7] mostrou que os operadores booleanos regularizados são fechados sobre eles e os implementou para modelos CSG, enquanto Desaulniers e Stewart [2] propuseram uma implementação para os modelos b-rep, utilizando um conjunto de operadores de Euler estendido.

A demanda por modelos mais gerais do que variedades bidimensionais ou r-conjuntos levou à proposta dos *NM*-conjuntos representáveis no espaço $\Re^3$ [6][8][10].

Observou-se que os algoritmos de operações booleanas conhecidos não conseguem manipular satisfatoriamente essa nova classe de objetos, principalmente no sentido de garantir a propriedade de fechamento.

Em 1991, Crocker e Reinke [1] sugeriram um algoritmo de operações booleanas para *NM*-conjuntos baseado no trabalho de Weiler [9]. Trata-se de um algoritmo heurístico com procedimentos redundantes afim de possibilitar a interatividade. Dados dois ou mais objetos, as entidades geométricas determinadas pelas suas interseções são referenciadas na estrutura topológica de cada um deles, em seguida são agrupadas em uma única estrutura (fusão) e posteriormente, são extraídas as entidades pertencentes ao objeto resultante da operação booleana (seleção). Duas deficiências foram detectadas:

- A propriedade de fechamento dos NM-conjuntos sobre as operações booleanas não é garantida.

- A realimentação visual intermediária foi pouco explorada, comprometendo a interatividade e levando o processo a cálculos desnecessários.

Este artigo propõe um algoritmo de operação booleana visando a:

- garantia da propriedade de fechamento dos *NM*-conjuntos sobre estas operações,
- interatividade e
- eficiência.

Em relação à propriedade de fechamento é apresentado um novo conjunto de operadores. Quanto à interatividade, propomos adiar o processo de fusão de entidades resultantes da interseção até o último passo do algoritmo, de forma que sejam evitadas operações desnecessárias quando se pretende fazer um "UNDO". Devido à complexidade das operações booleanas, achamos ainda interessante introduzir a possibilidade de interação com o usuário ao longo do processo, para que ele possa conduzir de forma mais eficiente o fluxo do algoritmo.

## 2 Terminologia e conceitos básicos

Para facilitar a leitura, apresentamos nesta seção conceitos básicos relacionados ao tema:

- **homeomorfismo**: Sejam $A$ e $B$ espaços métricos. Um homeomorfismo de $A$ sobre $B$ é uma bijeção contínua: $f : A \to B$ cuja inversa $f^{-1} : B \to A$ também é contínua. Diz-se, então, que $A$ e $B$ são homeomorfos ou equivalentes topologicamente [3]. Intuitivamente, pode ser pensado como uma transformação elástica que preserva a adjacência, ou seja, uma transformação sem cortes ou emendas [8].
- **disco aberto**: é uma porção do espaço bidimensional que contém um círculo de raio $\epsilon$ positivo, centrado em um ponto $P$ qualquer, excluindo-se os pontos sobre a circunferência [8].
- **fronteira**: a fronteira de um determinado objeto é formada pelos pontos cuja distância entre o objeto e o seu complemento é zero. Denota-se a fronteira de um conjunto $A$ por $b(A)$ [3].
- **fecho**: Dado um conjunto $A$, a operação $A \cup b(A)$ define a operação fecho sobre o conjunto $A$, denotada por $c(A)$ [7].
- **variedade bidimensional fechada** (*2D-manifold*): objeto para o qual todos os pontos sobre sua fronteira tem uma vizinhança topologicamente equivalente a um disco aberto [5].
- **função analítica**: uma função é analítica em um domínio se ela pode ser expandida em uma série de potência convergente sobre qualquer ponto do domínio [7].
- **conjunto semi-analítico**: um conjunto é semi-analítico se ele pode ser expresso como uma combinação booleana finita (expressa em termos dos operadores booleanos ordinários) dos conjuntos da forma: $\{(x,y,z) : F_i(x,y,z) \leq 0\}$, onde $F_i$ é analítica [7].
- **conjuntos regulares**: A regularização de um conjunto de pontos A, r(A), é definida por r(A) = c(i(A)), onde c(A) e i(A) denotam o fecho e o interior de A, respectivamente. Conjuntos que satisfazem a relação r(A)≡A são ditos regulares [7].
- **r-conjuntos** (*r-sets*): são subconjuntos do $\Re^3$ que sejam limitados, fechados, regulares e semi-analíticos. Um conjunto é limitado se ele ocupa uma porção finita do espaço, e é fechado quando contém suas fronteiras [7].
- **vértice-componente**: qualquer vértice isolado no $\Re^3$.
- **aresta-arame** (*wire-edge*): qualquer aresta não-adjacente a faces.

## 3 Espaço de modelagem

Requicha postula um espaço de modelagem cujos elementos são sólidos abstratos representáveis (r-conjuntos) e mostrou que estes elementos são fechados sobre operações booleanas regularizadas [7].

Os operadores booleanos do sistema GWB são regularizados e pressupõe que os objetos primitivos sejam variedades bidimensionais [4]. Entretanto, a utilização de operadores regularizados não garante que o resultado de uma operação booleana entre variedades bidimensionais seja também uma variedade bidimensional. A Figura 1 mostra que a união de dois cubos, que são variedades bidimensionais, pode gerar como resultado não uma variedade bidimensional, mas um r-conjunto.

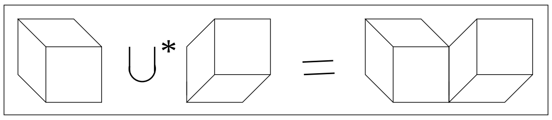

Figura 1: Operação booleana regularizada

Para que possamos representar uma gama maior de objetos a serem utilizados pelos operadores booleanos, introduzimos um espaço de modelagem $M$ para representação de objetos que sejam subconjuntos do espaço euclideano $\Re^3$, limitados, semi-analíticos e fechados, aos quais denominamos *NM*-conjuntos. Esta extensão visa preencher uma lacuna na representação de objetos geométricos no espaço tridimensional como mostra a Figura 2.

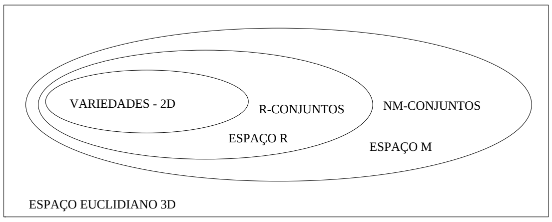


Figura 2: Espaço de modelagem

Considere o espaço de modelagem $R$ que cobre variedades bidimensionais fechadas e r-conjuntos. Alguns objetos que não poderiam ser representados no espaço $R$, são perfeitamente válidos no espaço $M$. O objeto representado na Figura 3 não se enquadra na definição de r-conjuntos, pois contém a aresta-arame **e** e a face **f**. Entretanto, ele está contido no espaço de modelagem $M$.

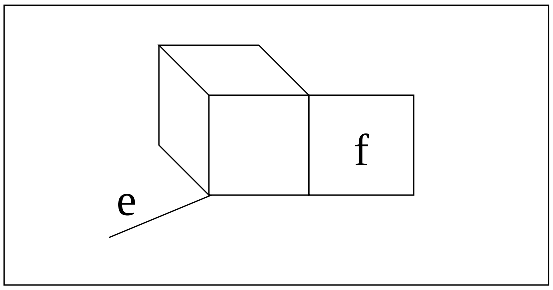


Figura 3: Exemplo de não-variedades

## 4 Um esquema de representação para os *NM*-conjuntos

Um esquema de representação para os *NM*-conjuntos pode ser considerado como um b-rep estendido, tendo como primitivas os vértices, as arestas, as faces, e os sólidos. Além disso, é interessante sob o ponto de vista da manutenção da consistência topológica distinguir os seguintes subconjuntos de primitivas [10]:

- **componente**: conjunto de entidades topológicas com conectividade maximal.
- **cadeia**: sequência ordenada de arestas. Pode ser fechada.
- **laço** (*loop*): fronteira de uma face. Um laço pode consistir apenas de um vértice ou um conjunto de cadeias e está associado a uma única face.
- **shell**: conjunto de faces conexas e orientadas coerentemente. Pode ser fechado.
- **cavidade**: contorno de um sólido. Uma cavidade consiste de um vértice, um conjunto de cadeias, um conjunto de shells ou a combinação dos dois últimos. Cada cavidade está associada a apenas um sólido.
- **ciclo topológico**: "orifício unidimensional". Um toro tem dois ciclos topológicos.
- **shell topológico**: "orifício bidimensional". Uma esfera com o interior vazio tem um shell topológico.

Estas primitivas se interrelacionam formando uma estrutura hierarquicamente organizada como mostra a Figura 4. Nesta organização cada entidade possui um identificador. Os objetos são formados pela combinação destas entidades. Os vértices, arestas, faces e sólidos representam as entidades mais elementares. Os sólidos são delimitados por cavidades, as faces por sequências de arestas (laços) orientadas e finalmente os vértices são os extremos das arestas. Este esquema é a base do módulo TDM (*Topological Data Module*), ou seja, Módulo de Dados Topológicos, sobre o qual são implementados os operadores booleanos [11].

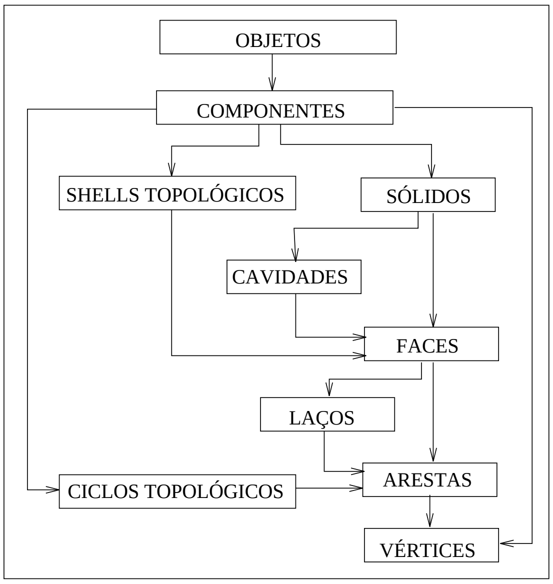


Figura 4: Módulo de Dados Topológicos (TDM)

O módulo TDM funciona como uma espécie de estrutura de dados "ativa", pois além de armazenar as entidades topológicas e suas relações de adjacências, provê um conjunto de mecanismos para verificar e manter a consistência dos dados. A manipulação dos dados da estrutura é realizada de forma transparente ao usuário através de funções organizadas em três grupos: funções de construção, funções de destruição e funções de consulta. Para cada função de construção existe uma função dual de destruição que realiza a operação inversa.

### 4.1 Funções de construção

O passo inicial no processo de construção de entidades topológicas de um objeto é a criação do próprio objeto na estrutura de dados do TDM. Esta tarefa é realizada pela função:

`TdOMakeObject(obj_id, regiao)`

onde `obj_id` é o identificador do objeto no TDM, e `regiao` identifica em qual região do espaço o objeto `obj_id` deve ser inserido. Em seguida, são montadas as entidades topológicas no TDM, respeitando as relações de adjacências entre elas.

**Exemplo**: Uma sequência de arestas é criada pelas funções:

`TdEMakeVertexComponent`

`TdEMakeEdgeVertex`

que inserem vértice-componente, e aresta e vértice a um objeto no TDM, respectivamente.Para fechar a sequência de arestas e criar uma face na estrutura utiliza-se a função:

`TdEMakeEdgeFace`

como mostra a Figura 5.

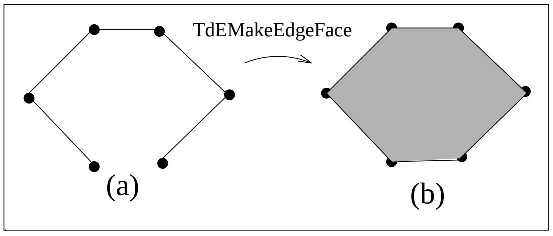


Figura 5: Exemplo de uma função de construção do TDM

### 4.2 Funções de destruição

Para cada função de construção existe uma função de destruição correspondente. Estas funções realizam operações inversas às realizadas pelas funções de construção.

**Exemplo**: Tomando-se a face do exemplo anterior, deseja-se eliminar um vértice e manter a face.

O TDM dispõe da função :

`TdVRKillVertexJoinEdge`

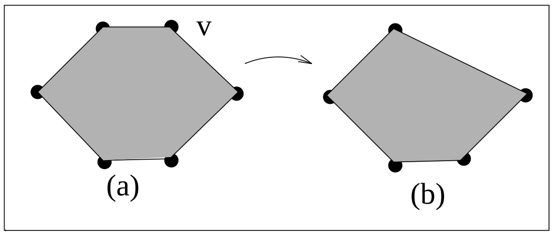


Figura 6: Exemplo de uma função de destruição do TDM

que elimina o vértice indicado, no caso **v** (Figura 6.a), e combina as duas arestas adjacentes a ele em uma única (Figura 6.b).

### 4.3 Funções de consulta

As informações sobre as entidades de um determinado objeto podem ser obtidas através das funções de consulta (*Inquiry*). Estas funções são bastante versáteis, pois com elas, pode-se obter informações acerca de qualquer entidade sem acessar diretamente a estrutura de dados.

**Exemplo**: A função:

`TdOAGetVertices(obj_id, v_qty, *vlist)`

retorna a quantidade de vértices (`v_qty`), e a lista de vértices (`*vlist`) de um objeto identificado por (`obj_id`).

## 5 Definição de um Novo Conjunto de Operadores Booleanos

A garantia do fechamento de uma operação booleana está intimamente ligada à definição do espaço de modelagem. Como vimos, os r-conjuntos são fechados sobre os operadores booleanos regularizados, enquanto que variedades bidimensionais não são.

Da mesma forma, a utilização de operadores booleanos ordinários em *NM*-conjuntos pode resultar em objetos inválidos para o espaço de modelagem $M$. Vejamos um caso pertinente. Dados os conjuntos $A$ e $B$ definidos por:

$$A = \{\{va_1, va_2, va_3, ..., va_i, ..., va_n\},\\ \{ea_1, ea_2, ea_3, ..., ea_j, ..., ea_p\},\\ \{fa_1, fa_2, fa_3, ..., fa_k, ..., fa_q\}\}$$

$$B = \{vb_1\}$$

onde $va_i$, $1 \leq i \leq n$, e $vb_1$ representam respectivamente os vértices do conjunto $A$, e o conjunto unitário $B$. Os elementos $ea_j$, $1 \leq j \leq p$, e $fa_k$, $1 \leq k \leq q$, são as arestas e as faces de $A$, respectivamente. As entidades definidas pertencem à fronteira de ambos os objetos. Supondo que o vértice $vb_1$ corresponde a um ponto $p$ e coincide com um ponto $pa$ do objeto $A$. Os resultados das operações de união, interseção e diferença entre $A$ e $B$ são os seguintes:

$A \cup B = A \cup \{vb_1\} = A \cup \{pa\} = A$
$A \cap B = \{pa\} \cap \{vb_1\} = \{vb_1\} = B$
$A - B = A - \{vb_1\} = A - \{pa\}$

Os resultados da união e interseção são triviais e válidos no espaço $M$. Já a operação diferença não preserva a propriedade do objeto ser fechado. Para contornar este problema, adotamos uma estratégia similar a de Requicha [7] e Masuda [6], no sentido de considerar o fecho do conjunto resultante. Assim, dados dois $NM$-conjuntos $X$ e $Y$, definimos como

- Operador booleano união:
  $X \cup^c Y = c(X \cup Y)$
  $= [i(X)-Y] \cup [i(Y)-X] \cup [i(X) \cap i(Y)] \cup [b(X)-Y] \cup [b(Y)-X] \cup [b(X) \cap b(Y)]$ (1)
- Operador booleano interseção:
  $X \cap^c Y = c(X \cap Y)$
  $= [i(X) \cap i(Y)] \cup [b(X) \cap i(Y)] \cup [b(Y) \cap i(X)] \cup [b(X) \cap b(Y)]$ (2)
- Operador booleano diferença:[1]
  $X -^c Y = c(X - Y)$
  $= [i(X) - Y)] \cup [b(Y) \cap i(X)] \cup [b(X) - Y] \cup [b(X) \cap b(Y)]$ (3)

  $Y -^c X = c(Y - X)$
  $= [i(Y)-X)] \cup [b(X) \cap i(Y)] \cup [b(Y)-X] \cup [b(Y) \cap b(X)]$ (4)

É fácil verificar que os NM-conjuntos são fechados sobre estes operadores booleanos. Voltando ao nosso exemplo, se considerarmos o fecho do resultado da operação diferença, tem-se $A -^c B = c(A - \{pa\}) = A$ pois, $pa \in b(A)$ e com isto, $pa \in A$, validando o conjunto para o espaço $M$.

Definidos os operadores booleanos, é fácil conceber um processo sistemático para selecionar dentre as entidades resultantes de uma interseção quais as válidas para a formação de um novo objeto, como mostraremos na próxima seção.

## 6 Implementação

O algoritmo que propomos para implementar as operações booleanas fechadas nos $NM$-conjuntos obedece o fluxograma mostrado na Figura 7.

A seguir detalhamos cada módulo do fluxograma.

### 6.1 Pré-processamento

O módulo de pré-processamento prepara os objetos para a operação booleana, destacando e posicionando-os convenientemente na cena através

[1] $X -^c Y \neq Y -^c X$, porque a diferença não é comutativa.

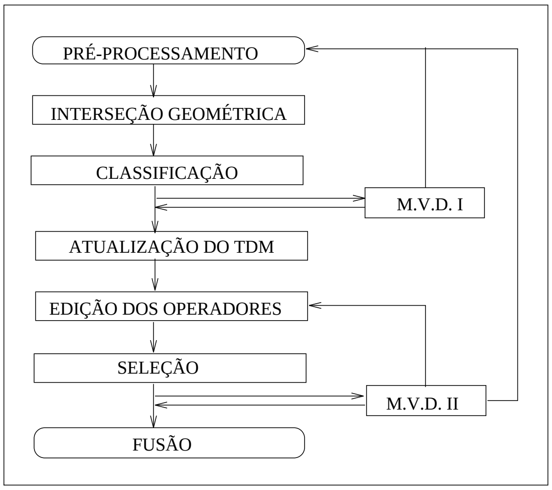


Figura 7: Fluxograma do algoritmo de Operação Booleana

das transformações geométricas disponíveis no sistema de modelagem. O algoritmo proposto processa somente dois objetos de cada vez.

Dentre as transformações geométricas, as mais requisitadas para o posicionamento dos objetos são as transformações de translação

### 6.2 Interseção geométrica e classificação

O módulo de interseção divide-se em duas partes, o cálculo das interseções geométricas entre as entidades referenciadas pela estrutura topológica e a classificação da entidades em relação aos dois objetos.

As interseções entre os objetos indicados no pré-processamento obedecem a seguinte sequência de testes entre as entidades que formam os objetos:

1. faces do objeto 1 **x** faces do objeto 2
2. faces do objeto 1 **x** arestas-arame do objeto 2
3. faces do objeto 1 **x** vértices-componente do objeto 2
4. arestas-arame do objeto 1 **x** arestas-arame do objeto 2
5. arestas-arame do objeto 1 **x** vértices-componente do objeto 2
6. vértices-componente do objeto 1 **x** vértices-componente do objeto 2

Sob o ponto de vista topológico, não há distinção entre formas curvilíneas e lineares. Um cubo, por exemplo, é homeomorfo a uma esfera! Entretanto, quando se trata de interseções entre duas formas geométricas, os algoritmos conhecidos consistem em resolver um sistema de equações que correspondem às representações algébricas das formas consideradas.

Em nossa implementação consideramos somente objetos poliédricos e portanto, as faces, as arestas e os vértices correspondem a polígonos planares, segmentos de reta e pontos, respectivamente.

Dos testes de interseção entre as entidades de cada objeto primitivo pode-se obter os seguintes resultados, quando existir:

1. numa interseção entre dois polígonos obtém-se polígnos ( Figura 8.a), segmentos(Figura 8.c,d) ou pontos(Figura 8.b);
2. numa interseção entre um polígono e um segmento obtém-se segmentos(Figura 9.b) ou pontos (Figura 9.a);
3. numa interseção entre um polígono e um ponto obtém-se um ponto(Figura 9.g);
4. numa interseção entre dois segmentos de reta obtém-se segmento(Figura 9.c) ou pontos (Figura 9.d);
5. numa interseção entre um segmento e ponto obtém-se um ponto(Figura 9.e);
6. numa interseção entre dois pontos obtém-se um ponto(Figura 9.f);

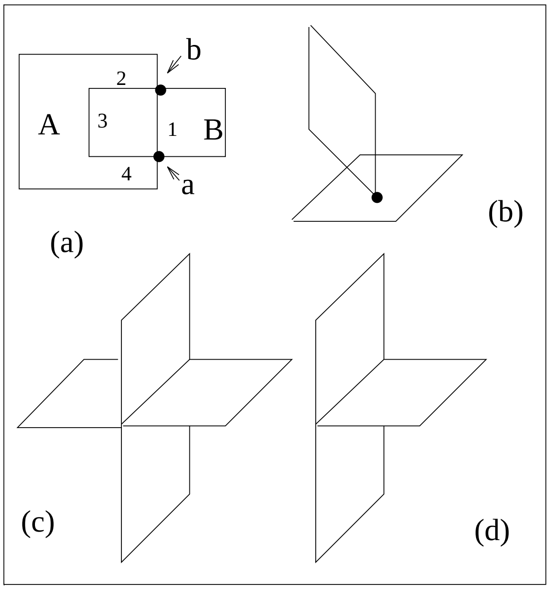


Figura 8: Interseções entre polígonos

Concomitantemente com as interseções, são realizadas as classificações das entidades dos dois *NM*-conjuntos, digamos $A$ e $B$, e as correspondentes entidades topológicas que resultam de interseções destes

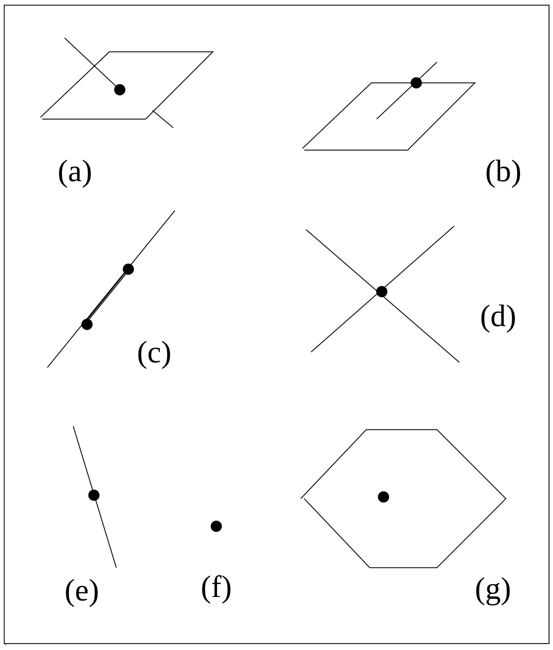


Figura 9: Interseções entre pontos, segmentos e polígonos

conjuntos. De acordo com as equações (1), (2), (3) e (4), os 8 grupos de entidades listadas na Tabela 1 são suficientes para caracterizar quaisquer das três operações booleanas. Consideramos como processo de classificação a atribuição de diferentes identificadores a cada grupo de entidades.

| GRUPO | | | IDENTIFICADOR |
|---|---|---|---|
| i(A) | ∩ | i(B) | 1 |
| i(A) | − | B | 2 |
| i(B) | − | A | 3 |
| b(A) | ∩ | i(B) | 4 |
| b(B) | ∩ | i(A) | 5 |
| b(A) | − | B | 6 |
| b(B) | − | A | 7 |
| b(A) | ∩ | b(B) | 8 |

Tabela 1: Classificação dos grupos de entidades

### 6.3 Atualização da estrutura topológica

Os resultados das interseções e classificação podem ser visualizados através do módulo de visualização I (M.V.D. I). Caso os resultados reflitam o esperado, a estrutura do TDM é atualizada com a inclusão dos resultados das interseções e classificação através do módulo de atualização topológica, que passa o controle do processo ao módulo de edição de operadores, senão o processo retorna ao pré-processamento para os ajustes necessários.

### 6.4 Edição de operadores

O módulo de edição indica a operação a ser realizada sobre as primitivas e passa esta informação ao módulo de seleção. Em uma operação booleana entre dois $NM$-conjuntos $A$ e $B$, os seguintes operadores estão disponíveis:

- União : $A \cup^c B$
- Interseção: $A \cap^c B$
- Diferença : $A -^c B$ ou $B -^c A$

### 6.5 Seleção das entidades topológicas

De acordo com a operação booleana indicada pelo módulo de edição de operadores e pelas expressões (1), (2), (3) e (4), são selecionados os grupos de entidades para compor o objeto resultante da operação:

- união ($A \cup^c B$): os grupos identificados por **1, 2, 3, 6, 7, 8** na Tabela 1;
- interseção ($A \cap^c B$): os grupos identificados por **1, 4, 5, 8** na Tabela 1;
- diferença ($A -^c B$): os grupos identificados por **2, 5, 6, 8** na Tabela 1;
- diferença ($B -^c A$): os grupos identificados por **3, 4, 7, 8** na Tabela 1;

Após realizada a seleção, um novo ponto de visualização permite observar os resultados (M.V.D. II) e tomar as decisões (confirmar a operação, re-editar os operadores, ou abortar e retornar ao início do processo). Caso deseje-se re-editar os operadores, o processo retorna ao passo de seleção e o procedimento é refeito. Se a decisão for abortar a operação, o módulo de visualização e decisão retira as entidades geradas na interseção das estruturas dos dois objetos.

### 6.6 Fusão

Se a decisão tomada no módulo M.V.D. II for confirmar a operação, as entidades selecionadas no passo anterior são agrupadas em um novo objeto e os objetos primitivos são retirados da estrutura, liberando a memória.

Este módulo finaliza o processo de operação booleana. O objeto resultante permanece armazenado no TDM, e pode ser utilizado como primitiva em uma outra operação booleana.

### 6.7 Módulos de visualização e decisão

Os módulos M.V.D. permitem a visualização de resultados intermediários e o controle do fluxo do algoritmo pelo usuário. A inclusão destes módulos é uma vantagem oferecida pelo nosso algoritmo. Diferente dos algoritmos que se tem conhecimento, este algoritmo permite a intervenção do usuário em pontos críticos do processo, afim de evitar cálculos desnecessários.

O M.V.D. I permite ao usuário observar os $NM$-conjuntos primitivos e entidades geométricas obtidas no processo de interseções. Isto possibilita tomar a decisão de continuar com a operação caso os resultados sejam os esperados e assim passar o controle da operação para o módulo de edição de operadores, ou senão parar o processamento caso os resultados da interseção não sejam satisfatórios, sem ter que prosseguir até o final da operação booleana.

O M.V.D. II possui os mesmos recursos do M.V.D. I, agora com relação aos resultados da seleção. Neste passo visualiza-se o objeto resultante de uma determinada operação booleana. Caso deseje-se confirmar este objeto como objeto final do processo, o fluxo de processamento prossegue para a fusão das entidades selecionadas em um $NM$-conjunto final. Por outro lado, o fluxo do algoritmo pode retornar ao módulo de edição para a escolha de uma nova operação ou retornar ao passo inicial.

## 7 Resultados

A Figura 10 ilustra o processo de operação booleana entre dois objetos maciços (sólidos), um cubo e um paralelepípedo.

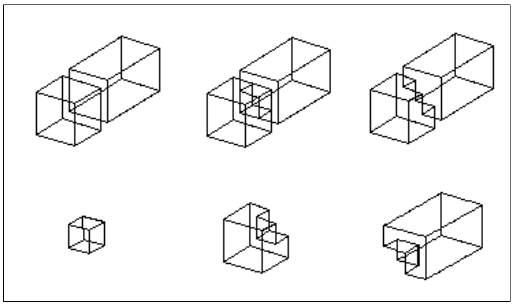

Figura 10: Interseção entre um cubo e um paralelepípedo

A Figura 11 ilustra o processo de operação booleana entre uma variedade bidimensional fechada (cubo) e um $NM$-conjunto (semi-plano).

Pode-se observar nos dois resultados mostrados: os objetos antes de serem processados, as linhas da interseção, a união, a interseção, e finalmente a diferença.

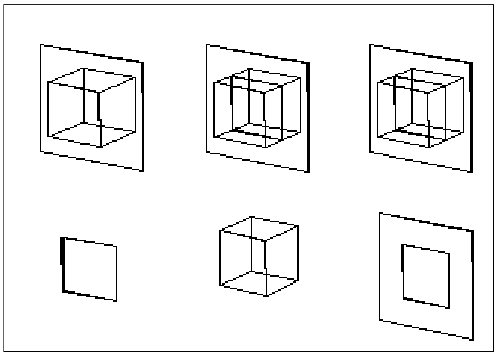

Figura 11: Interseção entre um cubo e um semi-plano

## 8 Conclusões

Este artigo apresenta um algoritmo de operação booleana fechada para um espaço de modelagem estendido em relação aos r-conjuntos, os *NM*-conjuntos. Através do adiamento do processo de fusão para o passo final do algoritmo a implementação da operação "UNDO" fica mais simples e o volume de operações envolvido fica reduzido, aumentando assim a interatividade. Além disso, com a participação ativa do usuário através dos módulos M.V.D. conseguimos obter um controle fino do processo.

Como continuidade deste trabalho pretendemos explorar a robustez do algoritmo implementado e concluir a implementação de algumas funções do TDM para suportar a implementação do nosso algoritmo.

## 9 Agradecimentos

Agradecemos aos revisores anônimos pelas críticas construtivas e ao CNPq como orgão financiador da Bolsa de Mestrado do primeiro autor.

## Referências

[1] G.A. Crocker and W.F. Reinke. An editable nonmanifold boundary representation. *IEEE Computer Graphics and Applications*, pages 39–51, mar 1991.

[2] H. Desaulniers and N. F. Stewart. An extension of manifold boundary representation to the r-sets. *ACM Transactions on Graphics*, 11(1):40–60, jan 1992.

[3] E.L. Lima. *Espaços métricos*. Impa e CNPq, Rio de Janeiro, 1988.

[4] M. Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press Inc., Maryland, 1988.

[5] M. Mäntylä. A note on modeling space of euler operators. *Computer Vision, Graphics and Image Processing*, 1(26):45–60, 1991.

[6] H. Masuda. Form-feature representation based on non-manifold geometric modeling. In *MICAD*, pages 17–35, Paris, França, fev 1992.

[7] A.A.G. Requicha. Representation of rigid solids - theory, methods and systems. *ACM Computer Surveys*, 12(4):437–364, 1980.

[8] K. Weiler. Edge-based data structures for solid modeling. *IEEE Computer Graphics and Applications*, 5(1):21–40, 1985.

[9] K. Weiler. *Topological structures for geometric modeling*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 1986.

[10] S.T. Wu. Non-manifold data models: Implementation issues. In *MICAD*, pages 37–56, Paris, França, fev 1992.

[11] S.T. Wu and L.G. Silveira Júnior. *Topological Data Module (TDM) Manual*. DCA/FEE/UNICAMP, Campinas - SP, Brasil, 1994. (em preparo).