Israel Felipe Prates

ESTUDO SOBRE PROPRIEDADES QUANTITATIVAS DE CRITÉRIOS DE SATISFAZIBILIDADE PROPOSICIONAL

(*versão de pré-defesa, compilada em 19 de dezembro de 2018*)

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Fabiano Silva.

Curitiba PR
2018

Resumo

Neste trabalho revisamos conceitos de lógica proposicional a fim de construirmos o conhecimento necessário para o estudo de operadores quantitativos. Estes operadores avaliam o resultado de uma fórmula pela quantidade de valores lógicos positivos (1 ou verdade) em sua entrada. Dois operadores quantitativos são estudados neste trabalho: o operador de particionamento lógico e o operador majoritário. Nossa proposta inclui a definição do operador de particionamento lógico, bem como conjecturas acerca da formulação de seus axiomas. Esperamos, por meio deste operador, prover um mecanismo capaz de utilizar a capacidade aritmética de computadores no contexto de satisfazibilidade.

Palavras-chave: satisfazibilidade, lógica proposicional, operador de particionamento, operador majoritário, propriedades lógicas quantitativas.

Abstract

In this work we review propositional logic concepts in order to build the knowledge necessary to study quantitative operators. These operators evaluate the result of a formula through the quantity of positive logic values (1 or true) in their input. Two quantitative operators are studied in this work: the logical partitioning operator and the majority operator. Our proposal includes the definition of the logical partitioning operator, as well as conjectures regarding its axioms. We hope, through this operator, to provide a mechanism capable of using the arithmetic capacity of computers in the context of satisfiability.

Keywords: satisfiability, propositional logic, partitioning operator, majority operator, quantitative logic properties.

Lista de Tabelas

Lista de Acrônimos

CNF	Conjunctive Normal Form
DNF	Disjunctive Normal Form
MNF	Majority Normal Form

Lista de Símbolos

$p \equiv q$	Equivalência entre p e q. Lê-se p equivale a q.
$p \wedge q$	Conjunção de p e q. Lê-se p e q.
$p \vee q$	Disjunção de p e q. Lê-se p ou q.
$\neg p$	Negação de p. Lê-se não p.
$p \rightarrow q$	Implicação de p e q. Lê-se p implica q.
$p \leftrightarrow q$	Dupla implicação de p e q. Lê-se p se, e somente se, q.
$p \vdash q$	Consequência lógica. Lê-se q é consequência lógica de p.
$\bot$	Falsum.
$\top$	Verum.

Sumário

1 Introdução

Conforme observado em Biere et al. (2009), a lógica está intrinsecamente ligada aos conceitos de validade e consistência que, de acordo com a forma como se aborda a lógica, podem demonstrar diferentes facetas. Quando vista pelo prisma de uma abordagem sintática, a lógica nos dá a teoria da prova. Neste contexto, todas as definições aplicáveis são restritas à estrutura gramatical das asserções, onde a validade é percebida como derivabilidade, ou a capacidade de se inferir uma asserção a partir de outras por meio de um conjunto de regras estabelecido. A derivabilidade é equivalente a se dizer que uma asserção q é provada a partir de asserções $p_1, p_2, \ldots, p_n$ (em linguagem simbólica $p_1, p_2, \ldots, p_n \vdash q$). De maneira similar, a consistência de um conjunto de asserções $\{p_1, p_2, \ldots, p_n\}$ é vista como a impossibilidade de se derivar uma contradição a partir deste. Por outro lado, a inconsistência é justamente identificada por meio de uma derivação onde se chega a uma contradição. Por fim, é possível definir estes conceitos em função um do outro: $p_1, p_2, \ldots, p_n \vdash q$ se, e somente se, $\{p_1, p_2, \ldots, p_n, \neg q\}$ é inconsistente.

Enquanto uma abordagem sintática nos faz perceber a lógica em termos de estrutura gramatical, uma abordagem semântica a interpreta em termos da correspondência entre um modelo e o mundo que se deseja representar (teoria dos modelos). A validade, portanto, é vista como a relação entre premissas e conclusões, tal que sempre que a premissa for verdade, a conclusão associada também é. Ainda do ponto de vista semântico, a consistência é vista como satisfazibilidade. Decorre que um conjunto de fórmulas é satisfazível quando existe um modelo capaz de tornar verdade todas as suas fórmulas.

Este trabalho se dedica a estudar operadores e propriedades de caráter quantitativo, isto é, que permitem a avaliação de expressões por meio da análise da distribuição da quantidade de valores lógicos positivos (verdade) presentes. A principal motivação para a elaboração deste estudo é prover os mecanismos que permitam utilizar a capacidade aritmética dos computadores no contexto de satisfazibilidade.

Este documento é organizado como segue: a lógica proposicional é definida no Capítulo 2, para que então o operador majoritário seja definido no Capítulo 3. Este é o primeiro operador neste documento a avaliar uma expressão lógica por meio da quantidade de valores lógicos positivos em sua entrada. No Capítulo 4 é apresentado o operador de particionamento lógico — também capaz de avaliar por quantidade — e são discutidas conjecturas acerca de suas propriedades (comutatividade, associatividade, distributividade e negação). A lógica proposicional é revisitada e é observado o caráter quantitativo de seus operadores fundamentais.

2 Fundamentos

Este capítulo tem por finalidade introduzir conceitos básicos e definições a serem referenciados no decorrer deste texto. O conteúdo a seguir é categorizado em quatro seções: Lógica Proposicional (Seção 2.1), Axiomas dos Operadores Fundamentais (Seção 2.2), Formas Normais Conjuntiva e Disjuntiva (Seção 2.3) e Satisfazibilidade Proposicional (Seção 2.4).

2.1 Lógica Proposicional

Conforme observado por Russell e Norvig (1995), esta lógica trata de asserções a respeito da realidade que se deseja descrever. Portanto, para que possamos definir formalmente uma proposição, é necessário definirmos antes o que é um valor lógico.

Seja $\mathbb{B}$ o conjunto $\{0, 1\}$: um valor lógico é uma constante b tal que $b \in \mathbb{B}$. Além disto, de modo a simplificar a escrita, podemos estabelecer as equivalências da Equação 2.1 para as constantes 0 e 1. Observe o uso do símbolo $\equiv$ para representar as relações de equivalência.

$$\begin{aligned} 0 &\equiv \text{Falsidade} \equiv \text{F} \equiv \bot \\ 1 &\equiv \text{Verdade} \equiv \text{V} \equiv \top \end{aligned} \tag{2.1}$$

Dando um passo adiante, uma proposição é uma afirmação arbitrária a respeito de uma realidade qualquer — afirmação esta que pode ser mapeada para um valor lógico. Portanto, pode ser vista como uma variável cujo domínio é $\mathbb{B}$, dita variável lógica. Chamamos valorada a variável lógica cujo valor subjacente é conhecido.

Operadores, em lógica proposicional, são funções f tais que:

$$f : \mathbb{B}^n \longmapsto \mathbb{B} \tag{2.2}$$

Onde $n \in \{1, 2\}$, dependendo da aridade do operador.

Neste trabalho, consideramos três como sendo os operadores fundamentais da lógica proposicional: conjunção, disjunção e negação, representados pelas funções *and*, *or* e *not*, respectivamente. Estas funções são definidas na Equação 2.3, onde *min* retorna o menor elemento dentre seus argumentos e max é a função análoga. Observe ainda que a lógica proposicional

pode ser definida em função de apenas dois destes operadores — por exemplo, por meio da disjunção e da negação.

$$\begin{aligned} and(p,q) &= min\{p,q\} \\ or(p,q) &= max\{p,q\} \\ not(p) &= 1 - p \end{aligned} \tag{2.3}$$

Os operadores podem, também, ser representados de maneira mais sucinta em sua forma simbólica, como na Equação 2.4.

$$\begin{aligned} and(p,q) &\equiv p \wedge q \\ or(p,q) &\equiv p \vee q \\ not(p) &\equiv \neg p \end{aligned} \tag{2.4}$$

Além destes três operadores básicos, podemos ainda definir outros, como a implicação e a equivalência (dupla implicação), representados pelas funções $implies$ e $equivalent$, definidas na Equação 2.5.

$$\begin{aligned} implies(p,q) &\equiv p \rightarrow q \equiv \neg p \vee q \\ equivalent(p,q) &\equiv p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p) \end{aligned} \tag{2.5}$$

Uma forma alternativa de se definir o comportamento dos operadores apresentados até então é por meio de suas tabelas verdade, conforme a Tabela 2.1.

Tabela 2.1: Tabela verdade dos operadores em lógica proposicional.

p	q	$p \wedge q$	$p \vee q$	$\neg p$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	F	F	V	V	V
F	V	F	V	V	V	F
V	F	F	V	F	F	F
V	V	V	V	F	V	V

Seguindo a definição dos operadores, podemos prosseguir para a próxima construção da lógica proposicional: a sentença. Uma sentença pode ser categorizada como atômica ou composta, onde uma sentença atômica (também chamada átomo ou literal) consiste de apenas uma constante ou proposição. Uma sentença composta, por sua vez, se enquadra em um dos seguintes casos:

- Duas sentenças conectadas por meio de operadores binários, também chamados conectivos lógicos
- Aplicação do operador unário de negação sobre uma sentença
- Uma sentença envolvida em parênteses

Esta definição de sentença nos leva à gramática livre de contexto (2.6), que estabelece a sintaxe de fórmulas na lógica proposicional:

$$\begin{aligned} S &\to A \mid C \\ A &\to \bot \mid \top \mid p \\ C &\to S\ N\ S \mid \neg S \mid (S) \\ N &\to \wedge \mid \vee \mid \to \mid \leftrightarrow \end{aligned} \tag{2.6}$$

É de se observar que a gramática (2.6) é ambígua, pois a expressão $P \wedge Q \vee R$ poderia ser avaliada tanto como $(P \wedge Q) \vee R$, quanto como $P \wedge (Q \vee R)$. De modo a desfazer essa condição de ambiguidade é necessário estabelecer uma ordem de precedência entre os operadores, conforme a Tabela 2.2, onde o operador de negação tem maior prioridade sobre os demais.

Tabela 2.2: Ordem de precedência dos operadores em lógica proposicional.

Precedência	Operador
0	$\neg$
1	$\wedge$
2	$\vee$
3	$\to$
4	$\leftrightarrow$

A raiz de uma expressão é dada pelo último operador nesta a ser avaliado, ou ainda pela raiz da árvore sintática obtida através do reconhecimento de uma palavra da linguagem. A quantidade de níveis de uma expressão é dada pela quantidade de níveis de sua árvore de derivação. Observe que, para que estas definições façam sentido, devemos assumir aridade arbitrária para os operadores *and* e *or*.

Resta, então, definir o conceito de valoração de fórmulas: uma fórmula está totalmente valorada quando todas as suas variáveis possuem um valor associado. De maneira análoga, uma valoração parcial é aquela que faz conhecer o valor de ao menos uma das variáveis da fórmula — mas não todas.

2.2 Axiomas dos Operadores Fundamentais

Nesta seção definimos os axiomas de identidade, comutatividade, distributividade, associatividade e complemento da perspectiva dos operadores fundamentais da lógica proposicional. Todas as definições aqui apresentadas podem ser verificadas por meio das respectivas tabelas verdade. Estas tabelas, por sua vez, acompanham a definição dos axiomas a que se referem.

2.2.1 Identidade

Existe um elemento neutro das operações de disjunção e conjunção. Formalmente, $\forall x \in \mathbb{B}$, as equivalências $x \vee \bot \equiv x$ e $x \wedge \top \equiv x$ são verdade, conforme a Tabela 2.3.

Tabela 2.3: Tabela verdade do axioma de identidade em lógica proposicional.

x	$x \vee \mathrm{F}$	$x \wedge \mathrm{V}$
F	F	F
V	V	V

2.2.2 Comutatividade

As operações de disjunção e conjunção independem da ordem de seus operandos. Formalmente, $\forall x, y \in \mathbb{B}$, as equivalências $x \vee y \equiv y \vee x$ e $x \wedge y \equiv y \wedge x$ são verdade, conforme demonstrado na Tabela 2.4.

Tabela 2.4: Tabela verdade do axioma de comutatividade em lógica proposicional.

x	y	$x \vee y$	$y \vee x$	$x \wedge y$	$y \wedge x$
F	F	F	F	F	F
F	V	V	V	F	F
V	F	V	V	F	F
V	V	V	V	V	V

2.2.3 Distributividade

A operação de disjunção pode ser distribuída sobre a conjunção. De maneira análoga, a conjunção pode ser distribuída sobre a disjunção. Formalmente, $\forall x, y, z \in \mathbb{B}$, as equivalências $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$ e $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$ são verdade, conforme a Tabela 2.5.

Tabela 2.5: Tabela verdade do axioma de distributividade em lógica proposicional.

x	y	z	$x \vee (y \wedge z)$	$(x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z)$	$(x \wedge y) \vee (x \wedge z)$
F	F	F	F	F	F	F
F	F	V	F	F	F	F
F	V	F	F	F	F	F
F	V	V	V	V	F	F
V	F	F	V	V	F	F
V	F	V	V	V	V	V
V	V	F	V	V	V	V
V	V	V	V	V	V	V

2.2.4 Associatividade

A ordem de avaliação de encadeamentos de operadores do mesmo tipo (disjunção ou conjunção) é irrelevante para a avaliação da fórmula. Formalmente, $\forall x, y, z \in \mathbb{B}$, as equivalências $(x \vee y) \vee z \equiv x \vee (y \vee z)$ e $(x \wedge y) \wedge z \equiv x \wedge (y \wedge z)$ são verdade, conforme a Tabela 2.6.

Tabela 2.6: Tabela verdade do axioma de associatividade em lógica proposicional.

x	y	z	$(x \vee y) \vee z$	$x \vee (y \vee z)$	$(x \wedge y) \wedge z$	$x \wedge (y \wedge z)$
F	F	F	F	F	F	F
F	F	V	V	V	F	F
F	V	F	V	V	F	F
F	V	V	V	V	F	F
V	F	F	V	V	F	F
V	F	V	V	V	F	F
V	V	F	V	V	F	F
V	V	V	V	V	V	V

2.2.5 Complemento

A disjunção entre uma variável lógica e seu complemento é uma tautologia, i.e., é sempre verdade. Analogamente, a conjunção de uma variável e seu complemento é uma contradição, i.e., é sempre falsidade. Formalmente, $\forall x \in \mathbb{B}$, as equivalências $x \vee \neg x \equiv \top$ e $x \wedge \neg x \equiv \bot$ são verdade, conforme demonstrado na Tabela 2.7.

Tabela 2.7: Tabela verdade do axioma de complemento em lógica proposicional.

x	$\neg x$	$x \vee \neg x$	$x \wedge \neg x$
F	V	V	F
V	F	V	F

2.3 Formas Normais Conjuntiva e Disjuntiva

Uma fórmula em lógica proposicional é dita estar na forma normal conjuntiva (CNF) quando esta é expressada por uma conjunção de disjunções, resultando em uma representação de dois níveis. De maneira análoga, uma expressão na forma normal disjuntiva (DNF) é uma disjunção de conjunções, também em dois níveis. A Equação 2.7 nos dá a forma de uma expressão em CNF, dado que uma cláusula é definida como a disjunção de um número arbitrário de átomos (negados ou não), seguindo a forma $v_1 \vee v_2 \vee \ldots \vee v_m$.

$$\begin{aligned} &C_1 \wedge C_2 \wedge \ldots \wedge C_n, \text{ onde} \\ &C_i \text{ é uma cláusula para todo } i \in [1..n] \end{aligned} \tag{2.7}$$

2.4 Satisfazibilidade Proposicional

Podemos classificar fórmulas da lógica proposicional em quatro categorias: válidas, inválidas, satisfazíveis e insatisfazíveis. Esta classificação ocorre em função de como se distribuem as linhas da tabela verdade da fórmula em questão, conforme a Tabela 2.8.

Tabela 2.8: Classificação de fórmulas em função da distribuição das linhas da tabela verdade.

Classificação	Distribuição
Válida	Todas as linhas tornam a fórmula verdadeira
Inválida	Ao menos uma linha torna a fórmula falsa
Satisfazível	Ao menos uma linha torna a fórmula verdadeira
Insatisfazível	Todas as linhas tornam a fórmula falsa

A satisfazibilidade booleana no contexto de lógica proposicional, ou simplesmente SAT, é o problema de se decidir se uma dada fórmula em lógica proposicional admite uma valoração que a torna verdadeira após avaliados todos os operadores presentes. Este problema pode ser visto ainda em termos sintáticos, onde um conjunto de fórmulas satisfazível é aquele do qual não se pode derivar φ e $\neg\varphi$ (conceito de consistência), onde φ é uma fórmula do conjunto satisfazível.

3 Trabalhos Relacionados

Este capítulo se dedica ao estudo de operadores majoritários e suas propriedades, cuja axiomatização foi apresentada e demonstrada em Amaru et al. (2016).

3.1 Funções Auto-Duais

Uma função lógica f, conforme definido em Sasao (2012), é dita auto-dual se $f(x_1, x_2, \ldots, x_n) = \neg f(\neg x_1, \neg x_2, \ldots, \neg x_n)$, onde $\neg$ é o operador de negação. Observe que a formulação $\neg f(x_1, x_2, \ldots, x_n) = f(\neg x_1, \neg x_2, \ldots, \neg x_n)$ é uma definição alternativa para o caráter auto-dual de uma função.

3.2 Função Majoritária

Uma função majoritária $M : \mathbb{B}^n \longmapsto \mathbb{B}$ de aridade ímpar n, denotada M_n, é definida como a função auto-dual que assume o valor verdade predominante em sua entrada. Observe que esta função é auto-dual pois, ao negar os valores da entrada, a predominância é invertida (se verdade era predominante, então falsidade se torna predominante). A função majoritária é definida formalmente na Equação 3.1.

$$M_n(x_1, x_2, \ldots, x_n) = \begin{cases} 1 \text{ se } \sum_{i=1}^{n} x_i \geq \left\lceil \frac{n}{2} \right\rceil \\ 0, \text{ caso contrário} \end{cases} \tag{3.1}$$

Em função de sua aridade, a forma mais natural de se representar o operador em questão é por meio de funções, de modo que não será adotada uma representação simbólica para o operador majoritário, em oposição aos operadores anteriormente apresentados.

3.3 Notação Vetorial de Variáveis Lógicas

Emprestando a definição de Amaru et al. (2016), seja v um arranjo de variáveis lógicas, o trecho de v indexado por $i \in [m..n]$ é denotado v_m^n. Observe que se $m > n$, então o intervalo de indexação é vazio e, portanto, indicativo de um trecho também vazio. Referências a um

elemento de v indexado por i são da forma v_i. A negação de um vetor de variáveis lógicas v_m^n, dada por $\neg v_m^n$, equivale a $\neg v_i; \forall i \in [m..n]$. Por fim, esta notação admite a supressão dos limites de indexação m e n, caso ambos sejam dispensáveis, indicando que todo o arranjo é referenciado.

De modo a complementar a definição acima, considere $\emptyset$ como sendo um sinônimo para um arranjo vazio. Ainda, $u \cup v$ representa a concatenação dos elementos de u e v. Analogamente, $u \setminus v$ representa todos os elementos em u exceto por aqueles que estão também em v.

3.4 Axiomatização do Operador Majoritário

Nesta seção discutimos o processo de axiomatização do operador majoritário. As formulações e provas de corretude dos axiomas estão disponíveis nas Subseções 3.4.1 e 3.4.2, respectivamente. Remetemos o leitor a Amaru et al. (2016) para acesso à prova de completude destes axiomas.

3.4.1 Formulação

Aqui é discutida a formulação dos axiomas do operador majoritário, conforme definido na Seção 3.2.

Comutatividade

Dado um vetor de variáveis lógicas v, a ordem de seus elementos não altera o resultado do operador majoritário aplicado sobre v. Formalmente, pode ser expressado como na Equação 3.2, dado v_1^n, para todo $i, j \in [1..n]$.

$$M_n(v_1^{i-1}, v_i, v_{i+1}^{j-1}, v_j, v_{j+1}^n) \equiv M_n(v_1^{i-1}, v_j, v_{i+1}^{j-1}, v_i, v_{j+1}^n) \tag{3.2}$$

Conservação da Maioria

Dado um arranjo de variáveis lógicas, a remoção ou inclusão de um par de elementos cujos valores sejam complementares conserva o resultado da aplicação do operador majoritário inalterado. Esta afirmação equivale à Equação 3.3, dados v_1^n, e v_i e v_j tais que $v_i \neq v_j$. Observe que $v_i \neq v_j$ implica $i \neq j$.

$$\begin{aligned} M_n(v_1^n) &\equiv M_{n-2}(v_1^n \setminus (v_i, v_j)) \\ M_n(v_1^n) &\equiv M_{n+2}(v_1^n \cup (\bot, \top)) \end{aligned} \tag{3.3}$$

Associatividade

A troca de duas variáveis entre operadores majoritários aninhados M_n que compartilham $n-2$ variáveis não altera o resultado final. Isto equivale à Equação 3.4, dado um arranjo v_1^{n-2} e variáveis a, b e c.

$$M_n(v_1^{n-2}, a, M_n(v_1^{n-2}, b, c)) \equiv M_n(v_1^{n-2}, b, M_n(v_1^{n-2}, a, c)) \tag{3.4}$$

Distributividade

A mudança do nível das variáveis de um operador majoritário não altera o resultado final do operador. Observe que esta operação traz as variáveis mais externas (em níveis mais baixos da árvore que representa a operação) para cima, enquanto as variáveis em níveis mais altos são levadas para baixo. Esta afirmação é equivalente à Equação 3.5, dados u_1^{n-1} e v_1^n.

$$\begin{aligned} M_n(u_1^{n-1}, M_n(v_1^n)) &\equiv M_n(M_n(u_1^{n-1}, v_1), M_n(u_1^{n-1}, v_2), \ldots, M_n(u_1^{n-1}, v_{\lceil \frac{n}{2} \rceil}), v_{\lceil \frac{n}{2} \rceil+1}, v_{\lceil \frac{n}{2} \rceil+2}, \ldots, v_n) \\ &\equiv M_n(M_n(u_1^{n-1}, v_1), M_n(u_1^{n-1}, v_2), \ldots, M_n(u_1^{n-1}, v_{\lceil \frac{n}{2} \rceil+1}), v_{\lceil \frac{n}{2} \rceil+2}, v_{\lceil \frac{n}{2} \rceil+3}, \ldots, v_n) \\ &\vdots \\ &\equiv M_n(M_n(u_1^{n-1}, v_1), M_n(u_1^{n-1}, v_2), \ldots, M_n(u_1^{n-1}, v_n)) \end{aligned} \tag{3.5}$$

Propagação da Negação

A negação de um operador majoritário aplicado sobre um arranjo v equivale ao mesmo operador aplicado sobre a negação do arranjo v. Esta afirmação é equivalente à Equação 3.6, dado um vetor de entrada v_1^n.

$$\neg M_n(v_1^n) \equiv M_n(\neg v_1^n) \tag{3.6}$$

3.4.2 Provas de Corretude

Comutatividade

Prova. Pela definição da função M_n, na Seção 3.2, observamos que o resultado da operação depende apenas do teste em que se verifica se a soma dos operandos atinge ou supera o limiar $\lceil \frac{n}{2} \rceil$. Por sabermos que a operação de soma é comutativa para os números naturais é possível afirmar que, independente da ordem dos operandos, a soma dos argumentos de M_n será a mesma para um dado vetor de entrada. Assim, é possível afirmar que M_n é também comutativa. □

Conservação da Maioria

Prova. A base deste axioma é que, em um sistema binário de votação por maioria, dois votos complementares se anulam. Isto ocorre pois o acréscimo (ou a remoção) de variáveis complementares em pares altera o limiar e a contribuição para se atingir o referido limiar na mesma proporção — para cada duas variáveis adicionadas (ou removidas) o limiar cresce (diminui) em 1, ao passo que a contribuição também aumenta (diminui) em 1. □

Associatividade

Prova. Esta prova é dividida em três casos que englobam todo o espaço Booleano.

Caso 1: existe ao menos um par $(v_i, v_j) \in v_1^{n-2}$ tal que $v_i \neq v_j$. Este caso indica que é possível remover de v o par de elementos complementares v_i e v_j por meio da aplicação do axioma de Conservação da Maioria até que se atinja um caso dentre 2 ou 3.

Caso 2: todos os elementos de v_1^{n-2} são Falsidade. Para $n > 3$ a decisão não depende das variáveis a, b e c, estando fixada em Falsidade. Isto ocorre pois o número de ocorrências do valor lógico Falsidade em M_n é no mínimo $n - 2$, superior ou igual ao limiar $\lceil \frac{n}{2} \rceil$ para todo $n \geq 3$. Para $n = 3$ o operador colapsa em uma conjunção da forma $M_3(0, a, M_3(0, b, c))$, caso em que a associatividade já foi provada por Amarú et al. (2014a).

Caso 3: todos os elementos de v_1^{n-2} são Verdade. Para $n > 3$ a decisão não depende das variáveis a, b e c, estando fixada em Verdade. Isto ocorre pois o número de ocorrências do valor lógico Verdade em M_n é no mínimo $n-2$, superior ou igual ao limiar $\lceil \frac{n}{2} \rceil$ para todo $n \geq 3$. Para $n = 3$ o operador colapsa em uma disjunção da forma $M_3(1, a, M_3(1, b, c))$, caso em que a associatividade já foi provada por Amarú et al. (2014a).

□

Distributividade

Prova. Esta prova é dividida em três casos que englobam todo o espaço Booleano. Observe que, por definição de M, n é ímpar.

Caso 1: metade dos elementos de u_1^{n-1} é Falsidade, a outra metade é Verdade. Neste caso os elementos de u se anulam através de aplicações sucessivas do axioma de Conservação da Maioria, conforme Equações 3.7 e 3.8.

$$\begin{aligned} M_n(u_1^{n-1}, M_n(v_1^n)) &\equiv M_1(\emptyset, M_n(v_1^n)) \\ &\equiv M_1(M_n(v_1^n)) \\ &\equiv M_n(v_1^n) \end{aligned} \tag{3.7}$$

$$
\begin{aligned}
M_n(M_n(u_1^{n-1}, v_1), M_n(u_1^{n-1}, v_2), ..., M_n(u_1^{n-1}, v_n)) &\equiv M_n(M_1(\emptyset, v_1), M_1(\emptyset, v_2), ..., M_1(\emptyset, v_n)) \\
&\equiv M_n(M_1(v_1), M_1(v_2), ..., M_1(v_n)) \\
&\equiv M_n(v_1, v_2, ..., v_n) \\
&\equiv M_n(v_1^n)
\end{aligned} \tag{3.8}
$$

As Equações 3.7 e 3.8 demonstram que qualquer reorganização das variáveis conforme a formulação deste axioma resulta em um operador da forma $M_n(v_1^n)$, de acordo com o comportamento esperado.

Caso 2: ao menos $\lceil \frac{n}{2} \rceil$ elementos de u_1^{n-1} são Falsidade. Neste caso, por definição de M, é impossível que o limiar $\lceil \frac{n}{2} \rceil$ seja atingido através de uma aplicação da forma $M_n(u_1^{n-1}, a)$, onde a é um valor lógico qualquer. Deste modo a afirmação $M_n(u_1^{n-1}, a) \equiv \perp$ se sustenta. Isto ocorre pois o número máximo de ocorrências de Verdade em $u \cup \{a\}$, expresso por $\lfloor \frac{n}{2} \rfloor = n - \lceil \frac{n}{2} \rceil$, é estritamente menor que o limiar $\lceil \frac{n}{2} \rceil$, pois n é ímpar. Observe ainda que este axioma postula que no mínimo $\lceil \frac{n}{2} \rceil$ instâncias de u devem ser distribuídas sobre v por meio de operadores da forma $M_n(u_1^{n-1}, v_i)$, assim a aplicação $M_n(M_n(u_1^{n-1}, v_1), M_n(u_1^{n-1}, v_2), \ldots, M_n(u_1^{n-1}, v_{\lceil \frac{n}{2} \rceil}), v_{\lceil \frac{n}{2} \rceil + 1}, v_{\lceil \frac{n}{2} \rceil + 2}, \ldots, v_n)$ equivale a $M_n(\perp_1, \perp_2, \ldots, \perp_{\lceil \frac{n}{2} \rceil}, v_{\lceil \frac{n}{2} \rceil + 1}, v_{\lceil \frac{n}{2} \rceil + 2}, \ldots, v_n)$, que é sempre falsa — de acordo com o comportamento esperado.

Caso 3: ao menos $\lceil \frac{n}{2} \rceil$ elementos de u_1^{n-1} são Verdade. Este caso é análogo ao caso 2, pois o limiar representado pelo operador majoritário mais externo garantidamente será atingido graças à definição de M, uma vez que é sabida a existência de ao menos $\lceil \frac{n}{2} \rceil$ elementos cujos valores lógicos são verdade.

□

Propagação da Negação

Prova. Este axioma é um caso especial da definição de funções auto-duais apresentada na Seção 3.1. □

3.5 Propriedades do Operador Majoritário

3.5.1 Conjunção

A conjunção de n variáveis lógicas $x_1, x_2, \ldots, x_n$ equivale ao operador majoritário da forma $M_{2n-1}(x_1^n, y_1^{n-1})$ onde y é composto apenas por ocorrências do valor lógico 0. Isto ocorre

pois, em lógica proposicional, a conjunção só é verdadeira se todos os seus operandos forem também verdadeiros. Do ponto de vista do operador majoritário, a única configuração da entrada em que um operador majoritário da forma acima se torna verdade é quando todos os operandos da conjunção são verdadeiros, pois já existem $n-1$ valores falsos na entrada. Exemplo: $a \wedge b \wedge c \equiv M_5(a, b, c, 0, 0)$, para $n = 3$.

3.5.2 Disjunção

A disjunção de n variáveis lógicas $x_1, x_2, \ldots, x_n$ equivale ao operador majoritário da forma $M_{2n-1}(x_1^n, y_1^{n-1})$ onde y é composto apenas pelo valor lógico 1. Partindo de um argumento semelhante ao da conjunção, a disjunção de n variáveis assume verdade se ao menos um de seus operandos for verdadeiro. No contexto do operador majoritário, fornecer $n-1$ valores positivos (além dos operandos da disjunção) equivale a configurar o operador de tal forma que se ao menos um dos operandos for verdadeiro, então o limiar será atingido. Exemplo: $a \vee b \vee c \vee d \equiv M_7(a, b, c, d, 1, 1, 1)$, para $n = 4$.

3.5.3 Simplificação

Um operador majoritário pode ser simplificado através da remoção de um par de ocorrências da mesma variável em complemento — ou simplesmente a remoção de um par $(\bot, \top)$. Exemplo 1: $M_7(a, a, a, b, c, \neg a, c) \equiv M_5(a, a, b, c, c)$, pela remoção de uma ocorrência de a e $\neg a$. Exemplo 2: $M_3(a, 0, 1) \equiv M_1(a)$, pela remoção de 0 e 1.

3.6 Forma Normal Majoritária

Conforme definição em Amarú et al. (2014b), uma forma normal majoritária (MNF) é uma expressão de dois níveis composta por operadores majoritários aninhados. Observe que é possível converter uma conjunção (disjunção) de n variáveis para uma expressão majoritária de aridade $2n-1$ como na Equação 3.9. Desta definição segue que uma fórmula em MNF é capaz de representar tanto expressões em CNF quanto DNF.

$$\begin{aligned} u_1 \vee u_2 \vee \ldots \vee u_n &\equiv M_{2n-1}(u_1^n, \top_1, \top_2, \ldots, \top_{n-1}) \\ v_1 \wedge v_2 \wedge \ldots \wedge v_m &\equiv M_{2m-1}(v_1^m, \bot_1, \bot_2, \ldots, \bot_{m-1}) \end{aligned} \tag{3.9}$$

3.7 Expressão Majoritária

Uma expressão majoritária, definida em Chou et al. (2016), é uma conjunção de funções majoritárias da forma $M_\alpha(x_1^\alpha) \wedge M_\beta(y_1^\beta) \wedge \ldots \wedge M_\gamma(z_1^\gamma)$. Pela definição de MNF, disponível na Seção 3.6, uma expressão majoritária é também uma forma normal majoritária.

3.8 Satisfazibilidade Majoritária

Dada uma fórmula em MNF cuja raiz seja uma conjunção, também chamada expressão majoritária, detectar uma contradição é possível pelas propriedades a seguir:

Propriedade 1: Há uma contradição na fórmula se dois de seus operadores majoritários mais internos $M_m(w_1^{m'}, x_1^{m''})$ e $M_n(y_1^{n'}, z_1^{n''})$, onde $m = m' + m''$ e $n = n' + n''$, contêm entradas $w_1^{m'}$ e $y_1^{n'}$ tais que $m' \geq \lceil \frac{m}{2} \rceil$, $n' \geq \lceil \frac{n}{2} \rceil$ e $w = \neg y$.

Exemplo: $M_5(M_5(a, \neg b, \neg c, d, e), M_3(f, g, 1), M_3(\neg a, b, c), 0, 0)$ é uma contradição, pois o vetor $u_1^3 = (a, \neg b, \neg c)$ ocorre como u e $\neg u$ na expressão majoritária e, em ambas as situações, ocupa mais da metade da entrada do operador da ocorrência.

Propriedade 2: Seja $\mu(x)$ o número mínimo de variáveis em x que devem ser valoradas para que $M(x) \equiv \top$ e $V(x)$ o conjunto de variáveis em x, então há uma contradição na fórmula se:

(i) Dois de seus operadores majoritários mais internos $M_m(u_1^m)$ e $M_n(v_1^n)$ possuem entradas u e v tais que $V(u) = V(v)$ e todas as ocorrências de uma variável em v são complementares às ocorrências da mesma variável em u (observe que $\neg\neg x = x$).

(ii) As desigualdades $\mu(u) \geq \left\lceil \frac{|V(u)|}{2} \right\rceil$ e $\mu(v) \geq \left\lceil \frac{|V(v)|}{2} \right\rceil$ forem verdade.

Exemplo: $M_3(M_3(a, b, \neg c), M_7(\neg a, \neg b, \neg b, \neg b, c, c, c), 0)$ é uma contradição, pois $u = (a, b, \neg c)$ e $v = (\neg a, \neg b, \neg b, \neg b, c, c, c)$ produzem $V(u) = V(v) = \{a, b, c\}$, $\mu(u) = 2 \geq \left\lceil \frac{|V(u)|}{2} \right\rceil$ e $\mu(v) = 2 \geq \left\lceil \frac{|V(v)|}{2} \right\rceil$.

4 Operador de Particionamento Lógico

4.1 Definição

Definimos o operador de particionamento lógico a partir do operador de cardinalidade, conforme estudado em Oliveira (2017), como a função $\Phi : \mathbb{B}^n \longmapsto \mathbb{B}$ de aridade n. Seu comportamento é dado por um conjunto $X \in \{P_\bot, P_\top\}$, onde $\{P_\bot, P_\top\}$ é uma partição do intervalo $[0..n]$, de modo que $P_\bot$ é a parte que faz o operador assumir falsidade, enquanto $P_\top$ é a parte que o torna verdade. Se a soma dos elementos da entrada estiver em $P_\top$, então o operador assume o valor 1. De maneira análoga, se a soma estiver em $P_\bot$, então seu valor é 0. Observe que o complemento de $X = P_\top$ restrito ao intervalo $[0..n]$ (escrito $X^c \cap [0..n]$) é igual a $P_\bot$, pois $\{P_\bot, P_\top\}$ é uma partição de $[0..n]$. Reciprocamente, $X^c \cap [0..n] = P_\top$, para $X = P_\bot$.

Dado um conjunto X conforme definido acima, a aplicação do operador de particionamento sobre um arranjo v_1^n é denotada $\Phi_X(v_1^n)$. Este operador é definido formalmente na Equação 4.1.

$$\Phi_X(v_1^n) = \begin{cases} 1, \text{ se } \sum_{i=1}^n v_i \in X \\ 0, \text{ se } \sum_{i=1}^n v_i \in X^c \cap [0..n] \end{cases} \tag{4.1}$$

Note que apesar da definição do operador uma contradição é a partição trivial $\{P_\bot\} = \{[0..n]\}$. Reciprocamente, uma tautologia é dada pela partição trivial $\{P_\top\} = \{[0..n]\}$.

4.2 Conversões

Nesta seção é esquematizado o processo de conversão dos operadores fundamentais da lógica proposicional, bem como o majoritário, para o contexto de particionamento.

4.2.1 Lógica Proposicional

Considere o argumento que, em lógica proposicional, o operador $or(v_1^2)$ representa uma partição $\{P_\bot, P_\top\} = \{\{0\}, \{1, 2\}\}$ do conjunto $[0..2]$ tal que, se $v_1 + v_2 \in P_\bot$, então $or(v_1^2) = \bot$ e

se $v_1 + v_2 \in P_\top$, então $or(v_1^2) = \top$. Uma extensão deste argumento para aridades arbitrárias é apresentada na Equação 4.2.

$$\begin{aligned} P_\bot &= \{0\} \\ P_\top &= [1..n] \\ or(v_1^n) &= \begin{cases} 1, \text{ se } \sum_{i=1}^n v_i \in P_\top \\ 0, \text{ se } \sum_{i=1}^n v_i \in P_\bot \end{cases} \end{aligned} \tag{4.2}$$

Observe que a definição de $or(v_1^n)$ em função das partições $P_\bot$ e $P_\top$ é semelhante à definição de Φ_X. De fato, a equivalência $or(v_1^n) \equiv \Phi_{P_\top}(v_1^n)$ se sustenta.

De maneira análoga, o operador $and(v_1^n)$ pode ser definido como na Equação 4.3, resultando na equivalência $and(v_1^n) \equiv \Phi_{P'_\top}(v_1^n)$.

$$\begin{aligned} P'_\bot &= [0..n-1] \\ P'_\top &= \{n\} \\ and(v_1^n) &= \begin{cases} 1, \text{ se } \sum_{i=1}^n v_i \in P'_\top \\ 0, \text{ se } \sum_{i=1}^n v_i \in P'_\bot \end{cases} \end{aligned} \tag{4.3}$$

4.2.2 Operador majoritário

Pela definição do operador majoritário, disponível na Seção 3.2, observamos que o critério de sua avaliação como Verdade depende do limiar $\lceil \frac{n}{2} \rceil$ ser atingido pela soma dos operandos. Esta afirmação pode ser formulada em termos de partições através de $\{P_\bot, P_\top\} = \{[0..\lfloor \frac{n}{2} \rfloor], [\lceil \frac{n}{2} \rceil..n]\}$. Observe que $\lfloor \frac{n}{2} \rfloor \neq \lceil \frac{n}{2} \rceil$ pois n, por definição do operador majoritário, é ímpar. A Equação 4.4 apresenta a equivalência que permite a conversão.

$$\begin{aligned} P_\bot &= [0..\lfloor n/2 \rfloor] \\ P_\top &= [\lceil n/2 \rceil..n] \\ M_n(v_1^n) &\equiv \Phi_{P_\top}(v_1^n) \end{aligned} \tag{4.4}$$

4.3 Proposta de formulação dos axiomas

Nesta seção são apresentadas conjecturas acerca dos axiomas de comutatividade, distributividade e negação para o operador de particionamento. Uma proposta de formulação do axioma de associatividade acompanhada de um contraexemplo demonstra que este operador não é associativo.

4.3.1 Comutatividade

A Equação 4.5 tem a finalidade de formalizar a conjectura que dois elementos v_i e v_j arbitrários podem ser trocados sem influenciar o resultado da operação.

$$\Phi_X(v_1^{i-1}, v_i, v_{i+1}^{j-1}, v_j, v_{j+1}^{n}) \equiv \Phi_X(v_1^{i-1}, v_j, v_{i+1}^{j-1}, v_i, v_{j+1}^{n}) \tag{4.5}$$

4.3.2 Associatividade

A Equação 4.6 apresenta o comportamento esperado da associatividade no contexto deste operador. Em seguida, a Equação 4.7 apresenta um contraexemplo para a associatividade. Em linhas gerais, é demonstrado que a ordem em que os operandos são avaliados pode influenciar o resultado final.

$$\begin{aligned}
&\Phi_X(u_1^{i-1}, u_i, v_{i+1}^{m-1}, \Phi_X(v_1^{j-1}, v_j, v_{j+1}^{n})) \not\equiv \Phi_X(u_1^{i-1}, v_j, v_{i+1}^{m-1}, \Phi_X(v_1^{j-1}, u_i, v_{j+1}^{n})) \\
&\text{Pois } \exists X, n, u, v, i, j \text{ tal que} \\
&\Phi_X(u_1^{i-1}, u_i, v_{i+1}^{m-1}, \Phi_X(v_1^{j-1}, v_j, v_{j+1}^{n})) \neq \Phi_X(u_1^{i-1}, v_j, v_{i+1}^{m-1}, \Phi_X(v_1^{j-1}, u_i, v_{j+1}^{n}))
\end{aligned} \tag{4.6}$$

A saber $X = \{1, 2\}$, $n = 3$, $u = (1, 0)$, $v = (1, 1, 0)$, $i = 1$ e $j = 3$, conforme Equação 4.7.

$$\Phi_{\{1,2\}}(1, 0, \Phi_{\{1,2\}}(1, 1, 0)) \neq \Phi_{\{1,2\}}(0, 0, \Phi_{\{1,2\}}(1, 1, 1)) \tag{4.7}$$

4.3.3 Distributividade

A Equação 4.8 tem a finalidade de documentar a maneira pela qual uma operação Φ_X é distribuída sobre uma operação Φ_Y. A distribuição se dá pela aplicação de Φ_X sobre $u \cup v_i$ para todo $i \in [1..n]$. Este processo gera n aplicações distintas de Φ_X, sobre as quais é aplicado Φ_Y.

$$\Phi_X(u_1^{m-1}, \Phi_Y(v_1^{n})) \equiv \Phi_Y(\Phi_X(u_1^{m-1}, v_1), \Phi_X(u_1^{m-1}, v_2), \ldots, \Phi_X(u_1^{m-1}, v_n)) \tag{4.8}$$

4.3.4 Negação

Seja $\{P_\perp, P_\top\}$ uma partição de $[0..n]$, a negação de $\Phi_{P_\top}$ é dada pelo complemento de $P_\top$ restrito a $[0..n]$. Visto que $\{P_\perp, P_\top\}$ é uma partição de $[0..n]$, então o complemento

de $P_\top$ restrito a $[0..n]$ é $P_\bot$. Observe que $\neg\Phi_{P_\top}(v_1^n) \neq \Phi_{P_\top}(\neg v_1^n)$, pois $\exists P_\top, v, n$ tal que $\neg\Phi_{\{1\}}(\bot, \top) \neq \Phi_{\{1\}}(\neg\bot, \neg\top)$.

$$\neg\Phi_{P_\top}(v_1^n) \equiv \Phi_{P_\bot}(v_1^n) \tag{4.9}$$

5 Conclusão

5.1 Contribuição

Nossas contribuições através deste trabalho são a proposta de formalização do operador de particionamento lógico a partir do operador de cardinalidade e as conjecturas acerca de sua axiomatização. Além disto, o estudo de aspectos quantitativos no contexto de SAT também ocorre como uma das principais contribuições. Dentre os aspectos do operador tocados pelas conjecturas está a propriedade de associatividade, acompanhada de um contraexemplo que caracteriza a descoberta da não-associatividade do operador. Além disto, é sugerido um esquema de conversão dos operadores da lógica proposicional e do operador majoritário para o contexto do operador de particionamento.

Outras contribuições são a apresentação de uma breve introdução à lógica proposicional e ao operador majoritário, com informações a respeito dos axiomas dos operadores, propriedades e formas normais.

5.2 Trabalhos Futuros

O caminho mais natural a se seguir a partir desta etapa é formalizar a axiomatização do operador por meio de provas de corretude para cada uma das conjecturas aqui discutidas. Concomitante à elaboração das provas, é de interesse investigar a existência e complexidade de casos particulares de satisfazibilidade onde seja eficiente o uso do operador. Por fim, condicionadas à existência de casos eficientes, a proposta e implementação de um resolvedor que tenha como estrutura interna o operador de particionamento.

Referências

Amaru, L., Gaillardon, P.-E., Chattopadhyay, A. e De Micheli, G. (2016). A sound and complete axiomatization of majority-*n* logic. *IEEE Transactions on Computers*, 65(9):2889–2895.

Amarú, L., Gaillardon, P.-E. e De Micheli, G. (2014a). Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization. Em *Proceedings of the 51st Annual Design Automation Conference*, páginas 1–6. ACM.

Amarú, L., Gaillardon, P.-E. e De Micheli, G. (2014b). Majority logic representation and satisfiability. Relatório técnico, Integrated Systems Laboratory (LSI) – EPFL.

Biere, A., Heule, M. e van Maaren, H. (2009). *Handbook of satisfiability*, volume 185, páginas 3–6. IOS press.

Chou, Y.-M., Chen, Y.-C., Wang, C.-Y. e Huang, C.-Y. (2016). Majorsat: A sat solver to majority logic. Em *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*, páginas 480–485. IEEE.

Oliveira, R. T. d. (2017). Arco consistência generalizada em codificações sat relativas.

Russell, S. J. e Norvig, P. (1995). *Artificial intelligence: a modern approach*, páginas 166–174. Prentice-Hall.

Sasao, T. (2012). *Switching theory for logic synthesis*, páginas 93–95. Springer Science & Business Media.