Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

UnB-CIC: Uma classe em LaTeX para textos do Departamento de Ciência da Computação

José Marcos Leite

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.a Dr.a Claudia Nalon

Brasília
2020

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

UnB-CIC: Uma classe em LaTeX para textos do Departamento de Ciência da Computação

José Marcos Leite

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof.a Dr.a Claudia Nalon (Orientadora)
CIC/UnB

Prof. Dr. Donald Knuth
Stanford University

Dr. Leslie Lamport
Microsoft Research

Prof. Dr. Edison Ishikawa
Coordenador do Bacharelado em Ciência da Computação

Brasília, 24 de dezembro de 2020

Dedicatória

Eu dedico essa música a primeira garota que tá sentada ali na fila. Brigado!

Agradecimentos

Nos *agradecimentos*

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

O *resumo*

Palavras-chave: LaTeX, metodologia científica, trabalho de conclusão de curso

Abstract

O *abstract* é o resumo

Keywords: LaTeX, scientific method, thesis

Sumário

Lista de Figuras

Capítulo 1

Introdução

Lógica nos fornece ferramentas para criar e reconhecer argumentos válidos. Embora seja custoso formalizar problemas do mundo real para poder utilizar estas ferramentas, é desejável, principalmente em sistemas críticos, ter a certeza de que a solução aplicada está correta. Alguns exemplos onde isto é empregado são: verificação de *hardware*, verificação de programas, verificação de protocolos etc.

Formalmente, um argumento válido pode ser reescrito como prova de teorema.

Cientistas da computação são apaixonados por automação, então é natural que esforços para prova automática de teoremas sejam feitos.

Capítulo 2

Definicoes

Nesta seção apresentamos as definições básicas para o resto do texto.

2.1 Linguagem

Trabalharemos com a linguagem lógica modal K_n.

Definição 1 Seja $P = \{p, q, r, \dots\}$ um conjunto finito de símbolos proposicionais, $\mathcal{A} = \{1, 2, 3, \dots, n\}, n \in \mathbb{N}$. Definimos o conjunto de fórmulas $\mathcal{FBF}$ indutivamente.

- Se $\varphi \in \mathcal{P}$ então $\varphi \in \mathcal{FBF}$
- Se $\varphi \in \mathcal{FBF}$, $\psi \in \mathcal{FBF}$ e $a \in \mathcal{A}$, então $(\varphi \wedge \psi) \in \mathcal{FBF}$, $(\varphi \vee \psi) \in \mathcal{FBF}$, $(\varphi \to \psi) \in \mathcal{FBF}$, $\boxed{a}\varphi \in \mathcal{FBF}$, $\langle a \rangle\varphi \in \mathcal{FBF}$ e $\neg\varphi \in \mathcal{FBF}$

Definição 2 Denotamos por $\mathcal{LP}$ o conjunto de literais proposicionais e por $\mathcal{LM}$ o conjunto de literais modais. $\forall p \in \mathcal{P}, \forall a \in \mathcal{A}$, então $p \in \mathcal{LP}, \neg p \in \mathcal{LP}$, $\boxed{a}p \in \mathcal{LM}$, $\boxed{a}\neg p \in \mathcal{LM}$, $\langle a \rangle p \in \mathcal{LM}$, $\langle a \rangle\neg p \in \mathcal{LM}$

Definição 3 Uma cláusula proposicional é uma disjunção de literais proposicionais.

Seja $\Sigma = \{0, 1\}$, Σ^* é o conjunto de todas as cadeias formadas com elementos de Σ. Em particular, ϵ representa a cadeia vazia. Construiremos cadeias em Σ^* para codificar a posição de ocorrência de uma subfórmula em uma fórmula. Seja $inv\colon \{\text{positiva, negativa}\} \mapsto \{\text{positiva, negativa}\}$ tal que $inv(\text{positiva}) = \text{negativa}$ e $inv(\text{negativa}) = \text{positiva}$.

Definição 4 Definimos a polaridade de uma subfórmula pela função $pol\colon \mathcal{FBF} \times \mathcal{FBF} \times \Sigma^* \mapsto \{\text{positiva, negativa}\}$. Para $\varphi, \chi_1, \chi_2 \in \mathcal{FBF}, s \in \Sigma^*, a \in \mathcal{A}, val \in \{\text{positiva, negativa}\}$.

- $pol(\varphi, \varphi, \epsilon) = \text{positiva}$.
- Se $pol(\varphi, \chi_1 \vee \chi_2, s) = val$, então $pol(\varphi, \chi_1, s0) = pol(\varphi, \chi_2, s1) = val$
- Se $pol(\varphi, \chi_1 \wedge \chi_2, s) = val$, então $pol(\varphi, \chi_1, s0) = pol(\varphi, \chi_2, s1) = val$
- Se $pol(\varphi, \chi_1 \to \chi_2, s) = val$, então $pol(\varphi, \chi_1, s0) = inv(val)$ e $pol(\varphi, \chi_2, s1) = val$
- Se $pol(\varphi, \Diamond_a \chi_1, s) = val$, então $pol(\varphi, \chi_1, s0) = val$
- Se $pol(\varphi, \boxed{a}\, \chi_1, s) = val$, então $pol(\varphi, \chi_1, s0) = val$
- Se $pol(\varphi, \neg\chi_1, s) = val$, então $pol(\varphi, \chi_1, s0) = inv(val)$

Dizemos que a polaridade de χ_1 em φ na posição s é $pol(\varphi, \chi_1, s)$.

Definição 5 Definimos o nível modal de uma subfórmula pela função $mlevel\colon \mathcal{FBF} \times \mathcal{FBF} \times \Sigma^* \mapsto \mathbb{N}$. Para $\varphi, \chi_1, \chi_2 \in \mathcal{FBF}, s \in \Sigma^*, a \in \mathcal{A}, val \in \mathbb{N}$.

- $mlevel(\varphi, \varphi, \epsilon) = 0$.
- Se $mlevel(\varphi, \chi_1 \vee \chi_2, s) = val$ ou $mlevel(\varphi, \chi_1 \wedge \chi_2, s) = val$ ou $mlevel(\varphi, \chi_1 \to \chi_2, s) = val$, então $mlevel(\varphi, \chi_1, s0) = mlevel(\varphi, \chi_2, s1) = val$
- Se $mlevel(\varphi, \Diamond_a \chi_1, s) = val$ ou $mlevel(\varphi, \boxed{a}\, \chi_1, s) = val$, então $mlevel(\varphi, \chi_1, s0) = val + 1$
- Se $mlevel(\varphi, \neg\chi_1, s) = val$, então $mlevel(\varphi, \chi_1, s0) = val$

Dizemos que o nível modal de χ_1 em φ na posição s é $mlevel(\varphi, \psi, s)$.

A semântica para lógica modal proposicional é dada por estruturas de Kripke. Uma estrutura de Kripke M é da forma $M = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{A}|}, \pi)$, onde $\mathcal{W}$ é um conjunto de mundos possíveis, $w_0 \in \mathcal{W}$, $\pi : \mathcal{W} \times \mathcal{P} \to \{\mathbf{true}, \mathbf{false}\}$, $\mathcal{R}_a \subseteq \mathcal{W} \times \mathcal{W}$ para todo $a \in \mathcal{A}$. Dizemos que uma fórmula φ é satisfeita na lógica modal K no modelo M no mundo w se, e somente se, $\langle M, w\rangle \models \varphi$, conforme segue:

- $\langle M, w\rangle \models \varphi$, se e somente se $\varphi \in \mathcal{P}$ e $\pi(w, \varphi) = \mathbf{true}$
- $\langle M, w\rangle \models \neg\varphi$, se e somente se $\langle M, w\rangle \not\models \varphi$
- $\langle M, w\rangle \models (\varphi \wedge \psi)$, se e somente se $\langle M, w\rangle \models \varphi$ e $\langle M, w\rangle \models \psi$
- $\langle M, w\rangle \models (\varphi \vee \psi)$, se e somente se $\langle M, w\rangle \models \varphi$ ou $\langle M, w\rangle \models \psi$
- $\langle M, w\rangle \models (\varphi \to \psi)$, se e somente se $\langle M, w\rangle \not\models \varphi$ ou $\langle M, w\rangle \models \psi$
- $\langle M, w\rangle \models \Diamond_a \varphi$, se e somente se $\exists w', (w, w') \in \mathcal{R}_a, \langle M, w'\rangle \models \varphi$
- $\langle M, w\rangle \models \boxed{a}\varphi$, se e somente se $\forall w', (w, w') \in \mathcal{R}_a, \langle M, w'\rangle \models \varphi$

Uma fórmula φ é localmente satisfatível se existe um modelo M tal que $\langle M, w_0\rangle \models \varphi$. Uma formula φ é globalmente satisfatível se existe um modelo M tal que para todo $w \in \mathcal{W}$ temos que $\langle M, w\rangle \models \varphi$. Escrevemos $M \models \varphi$ se, e somente se, $\langle M, w_0\rangle \models \varphi$

Podemos reduzir o problema de satisfatibilidade global ao problema de satisfatibilidade local com a extensão da linguagem K pelo operador universal $\boxed{*}$. Seja $M = (\mathcal{W}, w_0, \mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{A}|}, \pi)$, $\langle M, w\rangle \models \boxed{*}\varphi$ se, e somente se, para todo $w' \in \mathcal{W}$, $\langle M, w\rangle \models \varphi$.

Definição 6 Uma fórmula está na forma normal negada caso seja formada somente por simbolos proposicionais, $\neg$, $\wedge$, $\vee$, $\boxed{a}$ e $\Diamond_a$ para $a \in \mathcal{A}$, e a negação só é aplicada a simbolos proposicionais.

É importante ressaltar se $\varphi \in \mathcal{FBF}$ que não está na forma normal negada pode ser reescrita como $\psi \in \mathcal{FBF}$ na forma normal negada com sua semântica seja preservada. Isto é, para todo $\langle M, w\rangle$, $\langle M, w\rangle \models \varphi$ se, e somente se, $\langle M, w\rangle \models \psi$.

2.2 Forma Normal em Camadas

O cálculo a ser apresentado utiliza uma outra linguagem chamada de Forma Normal Separada em Níveis Modais(SNF_{ml}).

Definição 7 Uma fórmula em SNF_{ml} é uma conjunção de cláusulas. Para $ml \in \mathbf{N} \cup \{*\}$ e $l_1, l_2 \in \mathcal{LP}$, cada cláusula está em um dos três formatos:

- $ml : c$, onde c é uma cláusula proposicional

$$\frac{\begin{array}{c} ml : (D \vee l) \\ ml' : (D' \vee \neg l) \end{array}}{\sigma(\{ml, ml'\}) : D \vee D'} \qquad \frac{\begin{array}{c} ml : (l_1 \to \boxed{a} l) \\ ml' : (l_2 \to \diamond_a \neg l) \end{array}}{\sigma(\{ml, ml'\}) : \neg l_1 \vee \neg l_2} \qquad \frac{\begin{array}{c} ml_1 : (l'_1 \to \boxed{a} l_1) \\ ml_2 : (l'_2 \to \boxed{a} \neg l_1) \\ ml_3 : (l'_3 \to \diamond_a l_2) \end{array}}{\sigma(\{ml_1, ml_2, ml_3\}) : \neg l'_1 \vee \neg l'_2 \vee \neg l'_3}$$

$$\frac{\begin{array}{c} ml_1 : (l'_1 \to \boxed{a} l_1) \\ \vdots \\ ml_m : (l'_m \to \boxed{a} \neg l_m) \\ ml_{m+1} : (l' \to \diamond_a \neg l) \\ ml_{m+2} : (l_1 \vee \ldots \vee l_m \vee l) \end{array}}{\begin{array}{c} ml : \neg l'_1 \vee \ldots \vee \neg l'_m \vee \neg l' \\ ml = \sigma(\{ml_1, \ldots, ml_{m+1}, ml_{m+2} - 1\}) \end{array}} \qquad \frac{\begin{array}{c} ml_1 : (l'_1 \to \boxed{a} l_1) \\ \vdots \\ ml_m : (l'_m \to \boxed{a} \neg l_m) \\ ml_{m+1} : (l' \to \diamond_a l) \\ ml_{m+2} : (l_1 \vee \ldots \vee l_m) \end{array}}{\begin{array}{c} ml : \neg l'_1 \vee \ldots \vee \neg l'_m \vee \neg l' \\ ml = \sigma(\{ml_1, \ldots, ml_{m+1}, ml_{m+2} - 1\}) \end{array}}$$

Figura 2.1: Regras de inferência

- $ml : l_1 \to \boxed{a} l_2$
- $ml : l_1 \to \diamond_a l_2$

A satisfatibilidade de uma fórmula em SNF_{ml} é definida a partir da satisfatibilidade de K_n. Sejam $ml : \varphi$ e $ml : \psi$ cláusulas SNF_{ml} e M um modelo na lógica K_n.

- $M \models * : \varphi$ se, e somente se, $M \models \boxed{*} \varphi$.
- $M \models (ml : \varphi) \wedge (ml : \psi)$ se, e somente se, $M \models ml : \varphi$ e $M \models ml : \psi$.
- $M \models ml : \varphi$ se, e somente se, para todo w tal que $depth(w) = ml$, temos $\langle M, w \rangle \models ml : \varphi$.

[1] descreve uma função de tradução de K_n para SNF_{ml} bem como prova que a tradução de uma fórmula mantém sua satisfatibilidade.

2.3 Regras de inferência

[1] introduziu o cálculo dedutivo baseado em resolução RES_{ml} para lógica K_n. Para lidar com $*$ as regras de inferência são simplificadas com o uso de uma função parcial de unificação $\sigma : P(\mathbf{N} \cup \{*\}) \mapsto \mathbf{N} \cup \{*\}$, tal que $\sigma(\{ml, *\}) = ml$, $\sigma(\{ml\}) = ml$ e indefinida caso contrário. As regras de inferência de RES_{ml} são apresentadas na Figura 2.1, onde $* - 1 = *$ e m pode ser 0. Essas regras só valem se o resultado da unificação for definido. [1] prova correção e corretude de todas as regras.

2.4 Algoritmo

A seguir, descrevemos o algoritmo implementado no KSP.

Algorithm 1: KSP-Proof-Search

Result: Satisfatibilidade da fórmula

1 preprocessamento-da-entrada;
2 tradução-para-SNF;
3 preprocessamento-de-clausulas;
4 $\Gamma^{lit} \leftarrow \bigcup \Gamma^{lit}_{ml}$;
5 **while** $\Gamma^{lit} \neq \emptyset$ **do**
6   **for** *todo nível modal ml* **do**
7         $clausula \leftarrow \text{given}(ml)$;
8   **if** *não redundante(clausula)* **then**
9             $\text{GEN1}(clausula, ml, ml - 1)$;
10             $\text{GEN3}(clausula, ml, ml - 1)$;
11             $\text{LRES}(clausula, ml, ml)$;
12             $\Lambda^{lit}_{ml} \leftarrow \Lambda^{lit}_{ml} \cup \{clausula\}$;
13   **end**
14         $\Gamma^{lit}_{ml} \leftarrow \Gamma^{lit}_{ml} \setminus \{clausula\}$;
15   **if** $0 : false \in \Gamma^{lit}_{0}$ **then**
16   **return** insatisfativel;
17   **end**
18         $\Gamma^{lit} \leftarrow \bigcup \Gamma^{lit}_{ml}$;
19   **end**
20   **return** satisfativel;
21 **end**

KSP utiliza uma variação de conjunto de suporte. Técnica que restringe os candidatos possíveis para resolução. [1] demostra que a variação é a correta e completa em SNF_{ml}.

As linhas 1-3 aplicam algumas regras de simplificação, traduzem a fórmula para linguagem SNF_{ml} e constrõem os conjuntos *usable* e de suporte. As Linhas 9-11 aplicam regras de inferências descritas em 2.1.

A função given, Linha 7, é responsável por escolher uma cláusula dentre todas as candidatas possíveis. KSP implementa cinco variações dessa função: Menor, Mais antiga, Mais nova, Mínima e Máxima; e o usuário pode escolher qual deseja utilizar. Cada nível modal é independente, possibilitando até estratégias diferentes em níveis diferentes. Naturalmente, a função given só considera as cláusulas do nível modal pedido.

Na variação menor é escolhida uma cláusula com o menor tamanho. Na variação mais antiga, é usada uma estrutura de dados fila. Em mais nova, é usada uma estrutura de dados pilha. Em mínima, é escolhida uma cláusula com o menor tamanho dentre as cláusulas com mínimo literal máximo. Em máxima, é feita escolha análoga a mínima mas dentre cláusulas com máximo literal máximo.

Neste trabalho propomos novos métodos para seleção de cláusulas e comparamos com os métodos previamente utilizados no KSP.

2.5 Grafo

Um grafo é um par um ordenado (V, E), onde $E \subseteq V \times V$, chamamos V o conjunto de vértices e E o conjunto de arestas.

2.6 Matroide

Seja X um conjunto de objetos e $I \subseteq 2^X$ o conjunto de conjuntos independentes tal que:

1. $\emptyset \in I$
2. $A \in I, B \subseteq A \implies B \in I$
3. Axioma do troco, $A \in I, B \in I, |B| > |A| \implies \exists x \in B \setminus A : A \cup \{x\} \in I$
4. Se $A \subseteq X$ e I e I' são conjuntos independentes maximais de A então $|I| = |I'|$

Então (X, I) é um matroide. O problema combinatório associado a ele é: Dada um função de peso $w(e) \geq 0 \ \forall e \in X$, encontre um subconjunto independente com maior soma de pesos possível.

Referências

[1] Nalon, Cláudia, Clare Dixon e Ullrich Hustadt: *Modal resolution: Proofs, layers, and refinements*. ACM Trans. Comput. Logic, 20(4), agosto 2019, ISSN 1529-3785. `https://doi.org/10.1145/3331448`. 5, 6

Referências

[1] Nalon, Cláudia, Clare Dixon e Ullrich Hustadt: *Modal resolution: Proofs, layers, and refinements*. ACM Trans. Comput. Logic, 20(4), agosto 2019, ISSN 1529-3785. `https://doi.org/10.1145/3331448`. 5, 6