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	Dr. Leslie Lamport
Stanford University	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

Lista de Tabelas

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

Definições

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 enumerável 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 $(\varphi \wedge \psi) \in \mathcal{FBF}$, $(\varphi \vee \psi) \in \mathcal{FBF}$, $(\varphi \to \psi) \in \mathcal{FBF}$, $\boxed{a}\varphi \in \mathcal{FBF}$, $\Diamond_a\varphi \in \mathcal{FBF}$ 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 $p \in \mathcal{LP}, \neg p \in \mathcal{LP}$, $\boxed{a}p \in \mathcal{LM}$, $\boxed{a}\neg p \in \mathcal{LM}$, $\Diamond_a p \in \mathcal{LM}$, $\Diamond_a\neg p \in \mathcal{LM}$

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, $w_0 \in \mathcal{W}$, $\pi : \mathcal{W} \times \mathcal{P} \to \{\textbf{true}, \textbf{false}\}$, $\mathcal{R}_a \subseteq \mathcal{W} \times \mathcal{W}$ 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 $\langle M, w\rangle \models \psi$
- $\langle M, w\rangle \models (\varphi \rightarrow \psi)$, 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 símbolos proposicionais, $\neg$, $\wedge$, $\vee$, $\boxed{a}$ e $\Diamond_a$ para $a \in \mathcal{A}$, e a negação só é aplicada a símbolos 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 semântica equivalente. 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 $ml \in \mathbb{N} \cup \{*\}$ e $l_1, l_2 \in \mathcal{LP}$, cada cláusula está em um dos três formatos:

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

- $ml : l_1 \rightarrow \boxed{a} l_2$
- $ml : l_1 \rightarrow \Diamond_a l_2$

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 $\langle M, w \rangle \models ml : \varphi$.

Uma função de tradução de K_n para SNF_{ml} bem como prova de que a tradução de uma fórmula preserva satisfatibilidade podem ser encontradas em [1].

Seja $\geq$ uma ordem total sobre os símbolos proposicionais. Extendemos esta ordem para os Literais da seguinte forma: Se $p \in \mathcal{P}$, então $\neg p \geq \neg p$ e $\neg p \geq p$; Se $p, q \in \mathcal{P}$ e $p \geq q$, então $p \geq \neg q$.

Definição 8 O literal l é máximo em $\varphi \in SNF_{ml}$ se e somente se l ocorre em φ e não há $l_2 \neq l$ em φ tal que $l_2 \geq l$.

Note que podemos escolher qualquer ordem sobre os símbolos proposionais, assim l pode ser máximo numa ordem e não ser máximo em outra ordem.

2.3 Regras de inferência

O cálculo dedutivo baseado em resolução RES_{ml} para lógica K_n foi descrito em [1]. Para simplificar a descrição das regras de inferência faremos uso de uma função parcial de unificação $\sigma\colon P(\mathbb{N} \cup \{*\}) \mapsto \mathbb{N} \cup \{*\}$, 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. Demostrações de correção e corretude podem ser encotradas em [1].

[LRES]

$$\frac{\begin{array}{c} ml : (D \vee l) \\ ml' : (D' \vee \neg l) \end{array}}{\sigma(\{ml, ml'\}) : D \vee D'}$$

[MRES]

$$\frac{\begin{array}{c} ml : (l_1 \to \boxed{a}\, l) \\ ml' : (l_2 \to \Diamond_a \neg l) \end{array}}{\sigma(\{ml, ml'\}) : \neg l_1 \vee \neg l_2}$$

[GEN2]

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

[GEN1]

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

[GEN3]

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

Figura 2.1: Regras de inferência

2.4 Algoritmo

A seguir, descrevemos o algoritmo implementado no KSP.

Algorithm 1: KSP-Proof-Search

Result: Satisfatibilidade da fórmula

preprocessamento-da-entrada;
tradução-para-SNF;
preprocessamento-de-clausulas;
$\Gamma^{lit} \leftarrow \bigcup \Gamma^{lit}_{ml}$;
while $\Gamma^{lit} \neq \emptyset$ **do**
 for *todo nível modal ml* **do**
  $clausula \leftarrow \text{given}(ml)$;
  if *não redundante(clausula)* **then**
   $\text{GEN1}(clausula, ml, ml-1)$;
   $\text{GEN3}(clausula, ml, ml-1)$;
   $\text{LRES}(clausula, ml, ml)$;
   $\Lambda^{lit}_{ml} \leftarrow \Lambda^{lit}_{ml} \cup \{clausula\}$;
  end
  $\Gamma^{lit}_{ml} \leftarrow \Gamma^{lit}_{ml} \setminus \{clausula\}$;
  if $0 : false \in \Gamma^{lit}_0$ **then**
   return insatisfatível;
  end
  $\Gamma^{lit} \leftarrow \bigcup \Gamma^{lit}_{ml}$;
 end
 return satisfatível;
end

KSP utiliza uma variação de conjunto de suporte, técnica que restringe os candidatos possíveis para resolução, e a demostração da correção e completude dessa extensão para SNF_{ml} pode ser encontrada em [1].

Na versão para lógica clássica de conjunto de suporte o conjunto de cláusulas Δ é particionado em dois conjuntos Γ, o conjunto de suporte ou não processado, e Λ, o conjunto *usable* ou processado. Cláusulas são selecionadas de Γ, e passam para Λ, para aplicação de regras de inferência, os resolventes são inseridos em Γ.

Na extensão para SNF_{ml}, onde as clásulas são rotuladas pelo nível modal, as clásulas de todo nível modal é particionado em três conjuntos Γ^{lit}_{ml}, Λ^{lit}_{ml} e Λ^{mod}_{ml}. Cláusulas proposicionais são particionadas em Γ^{lit}_{ml} e Λ^{lit}_{ml} como no caso em lógica clássica. Clásulas modais são armazenadas em Λ^{mod}_{ml}. Note que nenhuma regra de inferência descrita na Seção 2.3

produz novas cláusulas modais.

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 na Seção 2.3.

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

Na variação *menor*, é selecionada uma cláusula com o menor tamanho de Γ^{lit}_{ml}.

Na variação *mais antiga*, é salvo a ordem nas quais as cláusulas foram adicionadas à Γ^{lit}_{ml} e é selecionada a que foi adicionada antes de todas as outras.

Mais nova é análoga a *mais antiga*, mas é selecionada a que foi adicionada depois de todas as outras.

Em *mínima*, é escolhida uma cláusula com o menor tamanho dentre as cláusulas com o menor literal máximo em Γ^{lit}_{ml}.

Em *máxima*, é feita escolha análoga a *mínima* mas dentre cláusulas com o maior literal máximo em Γ^{lit}_{ml}.

Neste trabalho propomos novos métodos para seleção de cláusulas e comparamos com os métodos previamente utilizados no KSP. Para comparação usaremos o *benchmark* [2] contendo 9 famílias de fórmulas e para cada família 21 fórmulas satisfatíveis e 21 insatisfatíveis.

A Tabela 2.1 a baixo mostra a quantidade de fórmulas resolvidas em até dez segundos para cada família e cada uma das cinco estratégias disponíveis no KSP.

Os resultados indicam que o KSP é muito eficiente para a maioria das famílias independente da estratégia utilizada, mas muito pode ser melhorado em k_branch_n, k_branch_p, k_d4_n, k_ph_n, k_ph_p, k_poly_n e k_poly_p.

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. $\emptyset \in I$

—	*mais antiga*	*mais nova*	*mínima*	*máxima*	*menor*
k_branch_n	1	1	2	1	2
k_branch_p	2	1	3	1	3
k_d4_n	7	4	7	6	7
k_d4_p	21	21	21	21	21
k_dum_n	21	21	21	21	21
k_dum_p	21	21	21	21	21
k_grz_n	17	13	21	13	21
k_grz_p	21	21	21	21	21
k_lin_n	21	21	21	21	21
k_lin_p	21	21	21	21	21
k_path_n	21	21	21	21	21
k_path_p	21	21	21	21	21
k_ph_n	2	2	3	2	3
k_ph_p	2	2	3	2	3
k_poly_n	8	5	8	8	8
k_poly_p	8	6	9	8	9
k_t4p_n	21	21	21	21	21
k_t4p_p	21	21	21	21	21

Tabela 2.1: Fórmulas resolvidas em até 10 seg.

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.

Capítulo 3

Proposta de Solução

Neste capítulo propomos novos métodos de seleção de cláusula para o KSP.

3.1 Proposta

Grafo

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, 7

[2] Balsiger, Peter, Alain Heuerding e Stefan Schwendimann: *A benchmark method for the propositional modal logics k, kt, s4*. J. Autom. Reason., 24(3):297–317, abril 2000, ISSN 0168-7433. `https://doi.org/10.1023/A:1006249507577`. 8

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, 7

[2] Balsiger, Peter, Alain Heuerding e Stefan Schwendimann: *A benchmark method for the propositional modal logics k, kt, s4*. J. Autom. Reason., 24(3):297–317, abril 2000, ISSN 0168-7433. `https://doi.org/10.1023/A:1006249507577`. 8