UniRuy & Área 1 | Wyden
PROGRAMA DE ENGENHARIA DA COMPUTAÇÃO
TEORIA DE COMPILADORES

JOÃO VICTOR DE DEUS MARTINS

Teoria de Compiladores: PROPRIEDADES DE LINGUAGENS REGULARES

Salvador - Bahia - Brasil

2022

JOÃO VICTOR DE DEUS MARTINS

Teoria de Compiladores: PROPRIEDADES DE LINGUAGENS REGULARES

Trabalho Acadêmico elaborado junto ao programa de Engenharia UniRuy & Área 1 | Wyden, como requisito para obtenção de nota parcial da AV1 na disciplina Teoria de Compiladores no curso de Graduação em Engenharia da Computação, que tem como objetivo consolidar os tópicos do plano de ensino da disciplina.

Orientador: Prof. MSc. Heleno Cardoso

Salvador - Bahia - Brasil
2022

TERMO DE APROVAÇÃO

JOÃO VICTOR DE DEUS MARTINS

TEORIA DE COMPILADORES: PROPRIEDADES DE LINGUAGENS REGULARES

Trabalho Acadêmico aprovado como requisito para obtenção de nota parcial da AV1 na disciplina Teoria de Compiladores, UniRuy & Área 1 | Wyden, pela seguinte banca examinadora:

BANCA EXAMINADORA

Prof°. MSc°. Heleno Cardoso
Wyden

Salvador, 05 de Novembro de 2022

Dedico este trabalho acadêmico a todos que contribuíram direta ou indiretamente com minha formação acadêmica.

Agradecimentos

Primeiramente agradeço a Deus. Ele, sabe de todas as coisas, e através da sua infinita misericórdia, se fez presente em todos os momentos dessa trajetória, concedendo-me forças e saúde para continuar perseverante na minha caminhada.

E a todos aqueles que contribuíram direta ou indiretamente para a minha formação acadêmica.

"A educação tem raízes amargas, mas os seus frutos são doces".

Aristóteles.

Resumo

As propriedades de fechamento em linguagens regulares são definidas como certas operações de linguagem regular que garantem a saída de linguagem regular. Um encerramento refere-se a realizar alguma operação em uma linguagem, resultando em uma nova linguagem que é do mesmo tipo que o que foi originalmente operado, ou seja, uma linguagem regular. A linguagem é uma coleção de strings. É natural pensar em operações comuns de conjunto, como união e interseção, como uma forma de combinar linguagens. Obviamente, a coleção de todos os idiomas é fechada em qualquer operação comum. União/interseção/complemento/diferença de conjuntos de strings ainda produz um conjunto de strings.

Palavras-chaves: Linguagens regulares, Pertença, Equivalência, Finitude, Lema de bombeamento.

Abstract

Closing properties in regular languages are defined as certain regular language operations that guarantee regular language output. A closure refers to performing some operation on a language, resulting in a new language that is of the same type as the " " that was originally operated on, that is, a regular language. The language is a collection of strings. It is natural to think of common set operations, such as union and intersection, as a way of combining languages. Of course, the collection of all languages is closed in any common operation. Union/intersection/complement/difference of sets of strings still produces a set of strings.

Keywords: Regular languages, Belonging, Equivalence, Finitude, Pumping lemma.

Lista de abreviaturas e siglas

DFAs	- Deterministic Finite Automaton (Autômato Finito Determinístico).
CFL	- Pumping Lemma for Contextless Languages (Lema de Bombeando para Linguagens Sem Contexto).
NFAs	- Nondeteterministic Finite Automata (Autômatos finitos não determinísticos).
String	- Cadeias de caracteres que armazenam dados textuais.

Sumário

1 PROPRIEDADES DE LINGUAGENS REGULARES 10

1.1 Introdução 10

1.2 Execução/Método 10

1.2.1 Repositório de Pesquisa 10

1.2.2 String de Busca por Repositório 10

1.2.3 Artigos Selecionados 10

1.2.4 Resenha dos Artigos Selecionados 10

1.2.5 Perguntas e Respostas 14

1.3 Conclusão 14

Referências[1] 15

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 PROPRIEDADES DE LINGUAGENS REGULARES

1.1 Introdução

Mecanismos de plataforma cruzada para reconhecer linguagens não regulares existem fora dos mecanismos de expressão regular. No entanto, os mecanismos de expressão regular precisam seguir regras rígidas estabelecidas pela ciência da computação teórica para serem considerados formais. O Teorema de Kleene afirma que linguagens regulares são equivalentes a autômatos finitos. Isso significa que qualquer idioma reconhecido por um sistema automático é regular.

1.2 Execução/Método

Resenha desenvolvida através de analises de artigos científicos apurados.

1.2.1 Repositório de Pesquisa

Google Acadêmico

1.2.2 String de Busca por Repositório

"Regular Language Properties"

1.2.3 Artigos Selecionados

Using Language Inference to Verify Omega-Regular Properties

Specifying Properties of a Language with Regular Expressions

Regular language type inference with term rewriting

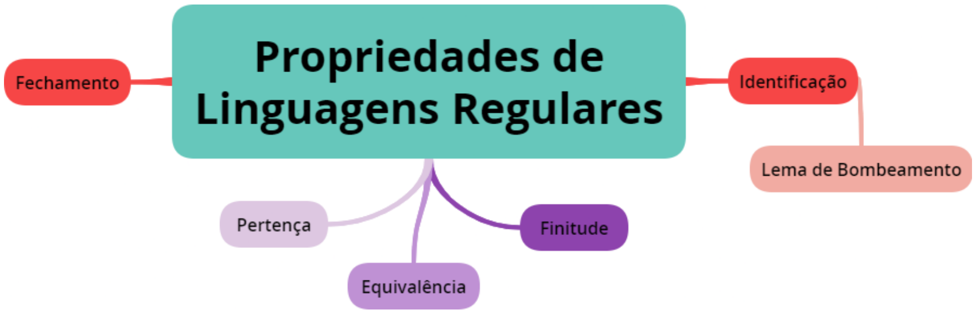
1.2.4 Resenha dos Artigos Selecionados

Definimos um encerramento como um conjunto de coisas. Descrevemos a propriedade fechada das linguagens regulares como operações implementadas em linguagens regulares que garantem a produção de novas linguagens regulares. Todos os idiomas regulares são desativados na operação acima. Essas operações são as seguintes: Klein Star Lock, União, interseção, cascata, Reabastecimento, revogação, diferença, homomorfismo, homomorfismo inverso. garante a integridade de sua estrela. Uma língua L1 é aquela que uma pessoa aprende em seu primeiro ano de vida, em Uma linguagem fechada requer Kleen L1* como sua palavra final. eu 1 Também são esperadas ocorrências consecutivas. O processo de fechamento das estrelas Kleen ocorre. Adicione suporte ao idioma L1 ao fechamento Kleen star. Tomamos essas medidas para atingir nossos objetivos. Crie uma representação literal da linguagem em uma máquina. Crie um novo estado inicial vinculando-o ao estado inicial original com uma transição nula. Crie um novo estado final adicionando uma transição nula entre ele e o estado final original. Em seguida, conecte o estado final original ao novo criando um novo estado final e renomeando o anterior para "estado final original". Do novo estado inicial para o novo estado final, crie uma transição nula. R é uma expressão regular na linguagem L. R+ é uma expressão regular na linguagem L+ , assumindo R = (a), então sua linguagem será L = a . Agora aplique o fechamento positivo em determinado regex e idioma, se R+ = (a) + então seu idioma será L+ = a, aa, aaa, aaaa .... Então eu+ ainda é uma linguagem regular. Portanto, os clusters positivos são satisfeitos.

O complemento da linguagem L é onde sigma mantém os símbolos de entrada usados para gerar a linguagem. Portanto, o complemento de uma linguagem regular é sempre regular. Sejam L1 e L2 linguagens de expressão regular R1 e R2, respectivamente. Então R1 + R2 (R1 U R2) também é uma expressão regular cuja linguagem é L3 = (R1 U R2). L3 também é uma linguagem regular Sejam L1 e L2 linguagens de expressão regular R1 e R2, respectivamente. Portanto, R1.R2 também é uma expressão regular e sua linguagem é L3 = (R1.R2). L3 também pertence à linguagem regular. Sejam L1 e L2 a linguagem das expressões regulares R1 e R2, respectivamente, cuja linguagem é a expressão regular da interseção de L1 e L2. Definir operador de diferença Sejam L1 e L2 a linguagem das expressões regulares R1 e R2 respectivamente, e uma linguagem seja a expressão regular de L1-L2. = string em L1, mas não em L2. Dada uma linguagem L, L é o conjunto de strings invertidos em L. Exemplo: L = 0, 01, 100 ; LR = 0, 10, 001 . LR ainda é uma linguagem regular É uma operação como a divisão. exemplo Seja L1 = 10, 100, 1010, 101110 L2 = 10 Vamos explicar o quociente esquerdo L1 / L2 (esquerda) = 10/10, 100/10, 1010/10, 101110/10 = e,0,10,1110 Vamos explicar o quociente correto L1 / L2 (direita) = 10/10, 100/10, 1010/10, 101110/10 = e, e, 10,1011 Então e, e, 10, 1011 obtido acima também é um idioma regular, pois autômatos finitos são possíveis. INIT também é conhecido como inicial ou prefixo. Se um idioma fornecer N strings, encontre o prefixo de cada string. Gera um novo idioma combinando os prefixos de todas as strings. A linguagem recém-gerada é a linguagem INIT, que ainda é uma linguagem regular. Ele é usado para substituir (substituir) um valor sigma por algum valor delta. Delta é apenas um símbolo. Delta contém alguns valores que substituem o valor sigma. Assumindo que "H"é delta, então podemos dizer H(L) = H(w) | w L exemplo Se L = 00, 101 e H (0) = "aa"e H ( ) = bb "H após substituição"L < TA H (L) = aaaa, bbaabb

É também a mesma técnica de substituição que o homomorfismo, mas a função de substituição é invertida. Substitua os valores delta por valores sigma. É representado por potências de -1, ou seja, (H-1). Dizemos que as duas expressões regulares R e S são São equivalentes se descrevem a mesma linguagem. Em outras palavras, se L(R) = L(S) para duas expressões regulares R e S então R = S. Existem dois lemas de bombeamento, que são definidos para 1. Idiomas regulares e 2. Contexto – Idiomas livres. Lema bombeamento para idiomas regulares para qualquer idioma regular L, existe um inteiro n tal que para todo x em L com |x| maior que n, existe um valor u, v, w em * tal que x = uvw e (1) |uv| é igual ou menor que n (2) |ui| são todos os inteiros i e ui são inteiros consecutivos). Isso significa que se uma sequência v for bombeada, ou seja, se for repetida várias vezes, a sequência resultante ainda permanece em L. Esse bombeamento de lema é usado como prova de irregularidades da linguagem. Uma linguagem deve conter pelo menos uma sequência de bombeamento que não esteja em L para ser julgada irregular. As linguagens regulares sempre têm uma lei à prova de bombeamento. Pumping Lemma nem sempre é preciso para determinar o estado de um idioma. Na verdade, manter o Pumping Lemma não garante que a linguagem seja regular. Por exemplo, vamos provar que L01 = 0n1n | n 0 é irregular. Vamos supor que L é regular, e então siga as regras acima por meio do lema de abstração. Agora, sejam x L e |x| não. Portanto, ao extrair o lema, existem u, v, w tais que (1) – (3) valem. Mostramos que para todo u, v, w, (1) – (3) não são verdadeiros. Se (1) e (2), mantenha x = 0n1n = uvw e |UV| n e | v | 1. Então você = 0uma, v = 0b, w = 0c1n onde: a + b n, b 1, c 0, a + b + c = n mas (3) falha porque i = 0 uv0com = uw = 0uma0c1n = 0a + c1n L porque a + c n. Pumping Lemma for Contextless Languages (CFL) O bombeamento de lema para CFL afirma que qualquer idioma sem regras pode ser bombeado várias vezes e ainda permanecer o mesmo idioma. Para fazer isso, dividimos qualquer idioma em cinco partes e bombeamos a segunda e a quarta string. Isso prova que uma linguagem não é CFL e pode ser usada como uma ferramenta para provar que uma linguagem não tem contexto. Uma linguagem é coloquialmente chamada de CFL se suas strings devem atender a critérios específicos para serem consideradas parte dela. Qualquer idioma que não cumpra essas regras não é CFL. É por isso que L tem um valor inteiro de n que pode ser encontrado em todo elemento x de L com comprimento x maior que n. Além disso, todo elemento v, w, x, y em * deve existir e satisfazer certos requisitos. Esses requisitos forçam todas as três partes da equação 1) $|vwx| <= n$, 2) $|wx| = 1$ e 3) $| 1 = 3$ para cada i de 1 a uuyeuxeuy.

Mapa Mental



1.2.5 Perguntas e Respostas

1. Quais as operações de uma linguagem regular?

Essas operações são as seguintes: Klein Star Lock, União, interseção, cascata, Reabastecimento, revogação, diferença, homomorfismo, homomorfismo inverso.

2. Defina encerramento de linguagens regulares?

operações implementadas em linguagens regulares que garantem a produção de novas linguagens regulares.

3. Quais os lemas de bombeamento?

1. Idiomas regulares e 2. Contexto – Idiomas livres.

1.3 Conclusão

A gramática regular e as expressões regulares aceitam linguagem, mas também podem criar novas linguagens. DFA e NFA são considerados aceitadores de linguagem, enquanto NFA são considerados criadores. Todas essas diferentes descrições de linguagem são igualmente poderosas porque todas são equivalentes.

Referências[1]

VARDHAN, A. et al. Using Language Inference to Verify Omega-Regular Properties. 2005.

TROUILLEUX, F. Specifying Properties of a Language with Regular Expressions. 3 abr. 2009.

HAUDEBOURG, T.; GENET, T.; JENSEN, T. Regular language type inference with term rewriting. 3 ago. 2020.

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.