Problema de la Satisfacibilidad Booleana de Concatenación de Rango Simple

Manuel Aguilera López M.AGUILERA@LAB.MATCOM.UH.CU
C311

Tutor(es): MSc. Fernando Rodríguez Flores

Resumen

El propósito de este trabajo es presentar una solución para un caso particular del problema de Satisfacibilidad Booleana, el problema de la Satisfacibilidad Booleana de Concatenación de Rango Simple. La solución a este problema se obtiene intersectando un lenguaje regular con una gramática de concatenación de rango simple y resolviendo el problema del vacío para la gramática obtenida, lo que es posible en tiempo polinomial. Este método generaliza otro caso particular del problema de Satisfacibilidad Booleana, el problema de Satisfacibilidad Booleana Libre del Contexto.

Abstract

The purpose of this paper is to present a solution to a particular case of the Boolean Satisfiability problem, the Simple Range Concatenation Satisfiability problem. This solution to this problem is obtained by intersecting a regular language with a simple range concatenation grammar and solving the emptiness problem in the resulting grammar, which is possible in a polynomial time. This method generalizes another particular case of the Boolean Satisfiability problem, the Context Free Boolean Satisfiability problem.

Palabras Clave: Problema de Satisfacibilidad Booleana, Gramática de concatenación de rango, Problema del vacío.
Tema: Teoría de Lenguajes Formales, Gramáticas de Concatenación de Rango.

1 Introducción

El problema de la satisfacibilidad booleana (en lo adelante SAT) consiste en determinar, dada una fórmula booleana, si existe una interpretación verdadera de la misma. Este problema en su forma general es NP-Completo, aunque existen casos particulares que se pueden resolver en tiempo polinomial como el 2-SAT y el Horn-SAT [1].

En [2] se presenta un nuevo caso particular del SAT soluble en tiempo polinomial: el problema de la Satisfacibilidad Booleana Libre del Contexto (en lo adelante SATCF) y tiene la característica adicional de que además de determinar si la fórmula es satisfacible o no, permite saber, en caso de lo que sea, cuántas interpretaciones tiene y cuáles son. Para lograr esto, se construye el lenguaje de todas las interpretaciones que hacen verdadera a la fórmula y se verifica si es vacío o no. Este lenguaje se puede construir a partir de la intersección de un lenguaje regular con un lenguaje libre del contexto. Ambos lenguajes tienen una complejidad polinomial con respecto al tamaño de la entrada, y la restricción que se le impone a las fórmulas es que el orden en que aparezcan las variables sea un orden "libre del contexto" [2].

En este trabajo se generaliza la idea utilizada en [2], pues en lugar de intersectar el lenguaje regular con un lenguaje libre del contexto, se intersecta con un lenguaje de concatenación de rango simple, que tiene un poder expresivo mayor que el de los libres del contexto y al mismo tiempo el problema del vacío se puede resolver en tiempo polinomial. Esto permite resolver instancias de SAT que no tienen un orden libre del contexto, en tiempo polinomial. La restricción que deben cumplir las fórmulas en este caso, es que el orden de las variables sea un orden "de concatenación de rango".

El objetivo de este trabajo es definir el problema de la Satisfacbilidad de Concatenación de Rango Simple (en lo adelante SATSRC), y proponer un algoritmo polinomial para resolverlo.

La estructura de este documento es la siguiente: en la Sección 2 se presentan definiciones necesarias para el planteamiento del problema y se describe la solución al SATCF. En la Sección 3 se definen las gramáticas de concatenación de rango

simple y se explica el algoritmo utilizado para resolver el problema del vacío para estas gramáticas. En la Sección 4 se define el SATSRC. En la Sección 5 se presenta la solución al SATSRC y un ejemplo. En la Sección 6 se presentan las conclusiones y las recomendaciones al trabajo.

2 Preliminares y Definiciones

En esta sección se presentan las definiciones necesarias para plantear formalmente el problema de la satisfacibilidad de concatenación de rango y su solución, además se describe la solución del problema de la satisfacibilidad libre del contexto.

Definición 1: Sea x_i una variable que ocurre en una fórmula lógica A; entonces cada una de sus ocurrencias en A se denomina **instancia de** la **variable** x_i.

Definición 2: Una **secuencia de instancias de variables** es una concatenación de instancias de variables.

Definición 3: La secuencia de instancias de variables que se obtiene de eliminar de una fórmula lógica todos los operadores y los paréntesis se denomina **Secuencia de Variables asociada a una Fórmula Lógica**.

Definición 4: Una **interpretación** de una fórmula lógica es una asignación de valores de verdad a cada una de sus variables. Los posibles valores de verdad son Verdadero o 1 y Falso o 0. Una interpretación de una fórmula *A*, determina un valor veritativo de *A*.

Definición 5: Una **fórmula** lógica es **satisfacible** si existe alguna interpretación se sus variables que la haga verdadera.

Definición 6: Problema de la Satisfacibilidad Booleana Libre del Contexto

El problema de la satisfacibilidad booleana libre del contexto consiste en determinar si existe alguna interpretación verdadera de una fórmula lógica con la restricción de que la secuencia de variables debe tener un orden libre del contexto. A continuación, se define "orden libre del contexto" y se describe el algoritmo propuesto en [2] para resolver el problema SATCF.

Definición 7: Una secuencia de variables tiene un **Orden Libre del Contexto** si se cumplen las siguientes condiciones:

1. Si hay una instancia de la variable x_j entre dos instancias consecutivas de la variable x_i, entonces todas las instancias de x_j deben aparecer entre esas mismas dos instancias de la variable x_i.

2. La condición 1 se cumple para todo par de variables x_i y x_j.

Para resolver un problema de la satisfacibilidad donde la secuencia de variables tiene un orden libre del contexto, se construye una representación polinomial del lenguaje de todas sus interpretaciones verdaderas, y se verifica si es vacío. Si lo es, la fórmula no es satisfacible, en caso contrario, la fórmula es satisfacible. Es posible determinar si un lenguaje libre del contexto es vacío o no utilizando el algoritmo descrito en [3].

El primer paso para obtener este lenguaje es construir un autómata booleano que reconoce el lenguaje de todas las interpretaciones verdaderas, suponiendo que todas las instancias de variables de la fórmula corresponden a **variables distintas**. Un autómata booleano es un autómata finito determinista [2], por lo que se puede afirmar que el lenguaje reconocido por el autómata es regular.

Una vez construido este autómata se procede a resolver el problema de que en la fórmula pueden existir dos o más instancias que correspondan a **variables iguales**, y por lo tanto, todas sus instancias deben tener el mismo valor. Para esto se le agrega al autómata una pila, con lo que se obtiene un autómata de pila. A partir del autómata de pila se obtiene una gramática libre del contexto que genera el mismo lenguaje. Entonces se resuelve el problema del vacío en esta nueva gramática libre del contexto.

En este caso particular del SAT, se restringe el orden de las variables a un orden libre del contexto para poder utilizar el poder de las gramáticas libres del contexto y resolver el problema del vacío en tiempo polinomial. Esta idea se puede generalizar si en lugar de usar lenguajes libres del contexto, se utilizan gramáticas con un poder expresivo mayor, y que permitan que el problema del vacío se solucione en tiempo polinomial. Las gramáticas de concatenación de rango simple, que serán presentadas en la siguiente sección, cumplen con estas condiciones.

3 Gramáticas de Concatenación de Rango Simple y Problema del Vacío

Las gramáticas de concatenación de rango simple son un caso particular de las gramáticas de concatenación de rango (RCG) [4].

Primeramente, se definen algunos de los elementos más importantes de las gramáticas de concatenación de rango:

Definición 8: Un **rango** es un par (i, j) de enteros no negativos que se denota por $\langle i..j \rangle$. Los rangos se utilizan para representar ocurrencias de sub-cadenas en una cadena dada.

Definición 9: Una **gramática de concatenación de rango** es un quíntuplo de la forma $G = (N, T, V, P, S)$ donde N es un conjunto finito de símbolos no terminales, T y V son conjuntos finitos disjuntos de símbolos terminales y variables respectivamente, $S \in N$ es el símbolo inicial y P es un conjunto finito de cláusulas:

$$\varphi_0 \rightarrow \varphi_1 \dots \varphi_m$$

Donde $m \geq 0$ y cada $\varphi_0, \varphi_1, \dots, \varphi_m$ es un predicado de la forma:

$$A(\alpha_1, \dots, \alpha_p)$$

donde $A \in N$. $p \geq 1$ es la aridad del predicado y cada argumento $\alpha_i, 1 \geq i \geq p$, es un elemento de $(T \cup V)^*$.

Definición 10: Un predicado cuyo no terminal es A se denota como A-predicado.

Definición 11: Una cláusula cuyo predicado en el lado izquierdo es un A-predicado, se llama A-cláusula.

Definición 12: La **sustitución de rango s**e define como un mecanismo de sustitución en el que se intercambia una variable por un rango. (También se conoce como atadura de rango).

La idea fundamental de las RCG es que los símbolos que ocurren en el argumento de una cláusula están atados a rangos de la cadena a través de un mecanismo de sustitución. De esta forma se pueden definir las derivaciones en una RCG.

Una derivación en una RCG consiste en la instanciación de las cláusulas cuyo predicado izquierdo es distinguido. Por cada una de estas cláusulas se buscan todas las posibles sustituciones de rangos a los símbolos del predicado izquierdo. Si existe alguna sustitución válida, entonces el predicado izquierdo de la cláusula se reemplaza por el predicado del lado derecho de la cláusula. Entonces se busca una cláusula cuyo predicado izquierdo sea uno de estos predicados derechos instanciados, y estas cláusulas se instancian a continuación. El proceso de derivación termina si se deriva ε (lo que significa que la cadena se reconoce) o cuando no se encuentra una cláusula que sirva como sustitución válida (y en este caso la cadena no se reconoce) [4] [5] [6].

Dentro de las RCG, las gramáticas de concatenación de rango simple son un subconjunto con propiedades atractivas, entre las que destaca que el problema del vacío se puede solucionar en tiempo polinomial, por lo que resultan especialmente convenientes para resolver problemas de SAT.

Definición 13: Una RCG G es ***simple*** si los argumentos en el lado derecho de una cláusula son variables distintas, y todas estas variables (y no otras) aparecen una sola vez en los argumentos del lado izquierdo.

En la solución del SATSRC es necesario resolver el problema del vacío para la intersección de un lenguaje regular con una gramática de concatenación de rango simple. Este problema es soluble en tiempo polinomial, y a continuación se presenta un algoritmo que lo resuelve [5].

En este algoritmo se asume que el lenguaje regular está definido a través de un autómata finito. Sean $B = (Q, E, f, q_0, F)$ un autómata finito de n estados, con $Q = \{q_0, q_1, \dots, q_n\}$ y $G = (N, T, V, P, S)$ una gramática de concatenación de rango simple con p producciones.

El algoritmo consiste en obtener, a partir de B y G, una gramática libre del contexto para el lenguaje de todas las posibles formas de generar la menor cadena en la intersección de los dos lenguajes. Esto es algo similar a lo que se hace en el proceso de reconocimiento de una cadena por una gramática de concatenación de rango [4] [5].

Los no terminales de esta gramática libre del contexto tendrán la forma $A((q_1, q_1'), \dots, (q_p, q_p'))$ para algún predicado $A(\alpha_1, \dots, \alpha_p) \in P$ de alguna cláusula de G, y $(q_1, q_1'), \dots, (q_p, q_p')$ son pares de estados de Q.

Las producciones de la gramática libre del contexto se forman de la siguiente forma: por cada cláusula

$$\varphi_0 \rightarrow \varphi_1 \dots \varphi_m,$$

se toman todas las posibles sustituciones de rango para las ocurrencias de terminales y variables en la cláusula, hacia pares de estados de Q, de forma que se cumplan las condiciones siguientes:

- si una ocurrencia de un terminal a se ata a un par de estados (q, q'), entonces debe existir una transición etiquetada "a" desde el estado q hacia q'.

- si variables y ocurrencias de terminales consecutivos en un argumento se atan a $(q_1, q_1'), (q_2, q_2') \dots, (q_t, q_t')$, para algún t, entonces $q_1' = q_2,\ q_2' = q_3, \dots, q_{t-1}' = q_t$.

Además, a la gramática se le añade una producción $S^+ \to S(q_0, q)$ para cada estado final $q \in V$ y el símbolo S^+ se convierte en el nuevo símbolo inicial.

Una vez obtenida la gramática libre del contexto se soluciona el problema del vacío sobre la misma, lo que es posible en tiempo polinomial [3]. Si esta gramática es vacía, entonces la intersección es vacía y si la gramática no es vacía, la intersección tampoco lo es.

4 Problema de la Satisfacibilidad Booleana de Concatenación de Rango Simple

En el SATCF, a la fórmula de entrada se le exige que sus instancias de variables tengan un orden libre del contexto. La solución al SATCF utiliza la intersección de un lenguaje regular con una gramática libre del contexto. En el caso del SATSRC se utilizará la intersección entre un lenguaje regular y una gramática de concatenación de rango simple.

La restricción aplicada a la fórmula de entrada en este caso consiste en que sus variables tengan un orden de concatenación de rango, que se define a continuación.

Definición 14: Una secuencia de instancias de variables de longitud n tiene un **orden de concatenación** de **rango simple**, si es posible construir una gramática de concatenación de rango simple, con un ***tamaño polinomial en n***, que genere el lenguaje de todas las cadenas $w_1 w_2 \dots w_n$, de longitud n sobre el alfabeto {0,1}, tal que para todo par de instancias de variables (x_i, x_j) que pertenezcan a la misma variable se cumple que $w_i = w_j$.

Una vez definido el problema SATSRC, y tomando en cuenta que el problema del vacío para la intersección de una gramática de concatenación de rango simple y un lenguaje regular se puede resolver en tiempo polinomial, es posible utilizar la idea presentada en [2] para resolverlo.

Si una fórmula booleana tiene un orden de concatenación de rango simple, entonces existe una gramática de concatenación de rango simple que genera el lenguaje de cadenas de sobre $\{0,1\}$ tales las instancias de variables de una misma variable tengan el mismo valor.

En este caso, el problema se puede solucionar en tiempo polinomial, como se muestra en la próxima sección.

5 Solución al SATSRC

En esta sección se presenta un algoritmo que permite resolver el SATSRC en tiempo polinomial, en este caso se les exige a las instancias de variables que cumplan con un orden de concatenación de rango simple.

Al igual que en el SATCF, se asume que todas las instancias de variables corresponden a variables distintas y se construye un autómata booleano que reconoce el lenguaje de todas las interpretaciones verdaderas bajo este supuesto.

Una vez obtenido el autómata booleano se intersecta con una gramática de concatenación de rango simple que genere cadenas de longitud n, donde todos los símbolos en posiciones que correspondan a instancias de una misma variable tengan el mismo valor. Luego se resuelve el problema del vacío de la forma explicada en la Sección 3. Si la intersección entre el lenguaje regular y la gramática de concatenación de rango es vacía, entonces la fórmula no es satisfacible, en caso contrario sí lo es.

Este problema del vacío se puede resolver en orden $O(n^k)$ donde n es el tamaño del autómata y k es la aridad de la gramática de concatenación de rango simple. El tamaño del autómata es $O(m^2)$ donde m es la cantidad de instancias de variables de la fórmula.

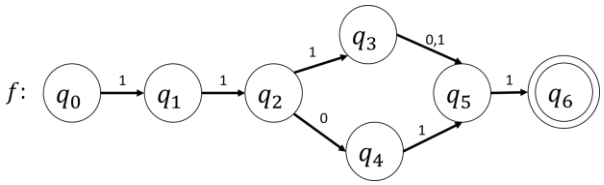
Esta complejidad se puede mejorar. Si la fórmula se plantea en forma normal conjuntiva, la complejidad del autómata es $O(m)$.

A continuación, se resuelve un ejemplo del SATSRC:

Ejemplo: Sea $A = (X_2 \wedge X_1) \wedge (X_3 \wedge X_2) \wedge X_1$ una fórmula booleana, se desea saber si A es satisfacible. La secuencia de variables de A es $X_2 X_1 X_3 X_2 X_1$. Las instancias de variables de esta fórmula presentan un orden de concatenación de rango simple que no cumple con un orden libre del contexto.

Sea $B = (Q, E, f, q_0, F)$ un autómata finito determinista que reconoce el lenguaje de las interpretaciones que hacen verdadera a A con

$\boldsymbol{Q} = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$, $\boldsymbol{q_0} = q_0$, $\boldsymbol{F} = \{q_6\}$, $\boldsymbol{E} = \{0,1\}$,



Sea $G = (N_0, T_0, V_0, P_0, S_0)$ una gramática de concatenación de rango simple que genera las cadenas en las que todos los símbolos en posiciones que corresponden a instancias de una misma variable tienen el mismo valor.

$\boldsymbol{N_0} = \{A\}$, $\boldsymbol{T_0} = \{0,1,\varepsilon\}$, $\boldsymbol{V_0} = \{X, Y\}$ $\boldsymbol{S_0} = S$,

$P_0 =$

$$S(XY) \to A(X,Y)$$
$$A(X0, Y0) \to A(X,Y)$$
$$A(X1, Y1) \to A(X,Y)$$
$$A(\varepsilon, 0) \to \varepsilon$$
$$A(\varepsilon, 1) \to \varepsilon$$

Al aplicar el algoritmo para resolver el problema del vacío sobre la intersección de A y B, la gramática libre del contexto $C = \{N_1, T_1, P_1, S_1\}$ resultante tiene la forma:

$P_1 =$

$$S^+(0,6) \to S(0,6)$$
$$S(0,6) \to A((0,3),(3,6))$$
$$S(0,6) \to A((0,4),(4,6))$$
$$A\big((0,3),(3,6)\big) \to A\big((1,3),(5,6)\big)$$
$$A\big((1,3),(5,6)\big) \to A\big((2,3),(6,6)\big)$$
$$A\big((2,3),(6,6)\big) \to \varepsilon$$
$$A\big((0,4),(4,6)\big) \to A\big((1,4),(5,6)\big)$$
$$A\big((1,4),(5,6)\big) \to A\big((2,4),(6,6)\big)$$
$$A\big((2,4),(6,6)\big) \to \varepsilon$$

Los no terminales de N_1 tienen la forma $A((q_1, q_1'), \dots, (q_p, q_p'))$ para algún predicado $A(\alpha_1, \dots, \alpha_p) \in P_0$ y $(q_1, q_1'), \dots, (q_p, q_p')$ son pares de estados de Q. Se define $S_1 = S^+$ y $T_1 = \varepsilon$.

Una vez obtenida C se resuelve el problema del vacío en tiempo polinomial [3], obteniéndose como resultado que C no es vacío y por lo tanto A es satisfacible.

6 Conclusiones y recomendaciones

En este trabajo se generalizó la propuesta realizada en [2], al contemplar otros tipos de órdenes, en particular el orden de concatenación de rango, y la gramática que lo genera cumple propiedades como que el problema del vacío es soluble en tiempo polinomial y tiene un poder expresivo mayor que las gramáticas libres del contexto.

Las definiciones y algoritmos utilizados permiten aplicar esta idea a cualquier tipo de lenguaje que genere órdenes no libres del contexto y que el problema del vacío sea soluble en tiempo polinomial.

El algoritmo propuesto tiene una complejidad $O(n^k)$ donde n es la longitud de la fórmula y k es la aridad de la gramática. Esto implica que las gramáticas utilizadas para describir el orden deben ser seleccionadas de manera que su aridad sea la menor posible.

Como recomendación se propone estudiar el algoritmo de intersección de lenguajes regulares con gramáticas de concatenación de rango simple para el caso en que el lenguaje regular es finito, donde la complejidad del algoritmo pudiera reducirse de manera notable.

Por otra parte, se pueden considerar otros tipos de formalismos, que no sean las gramáticas de concatenación de rango simple, siempre que el problema del vacío sea polinomial.

Finalmente, aunque el SATSRC, al igual que el SATCF, pueden no tener una aplicación práctica inmediata, su valor fundamental consiste en la forma novedosa de resolver el problema.

Referencias

[1] L. Kallmeyer, «Parsing Beyond Context-Free Grammars,» Springer, Tübingen.

[2] J. Hopcroft, J. Ullman y R. Motwani, «Introduction To Automata Theory, Language and Computation,» Addison-Wesley, 2001.

[3] J. y. o. Gu, «Algorithms for the satisfiability (SAT) problem: A survey . [aut libro] Dingzhu Du, Jun Gu y Panos Pardalos. Satisfiability Problem: Theroy and Applications.,» American Mathematical Society, 1996.

[4] A. Fernández Arias, «El Problema de la Satisfacibilidad Booleana Libre del Contexto. Un Algoritmo polinomial.,» La Habana, 2007.

[5] P. Boullier, «Proposal for a Natural Language Processing Syntactic Backbone,» INRIA, Paris, 1998.

[6] E. Bertsch y M.-J. Nederhof, «On The Complexity Of Some Extensions Of RCG Parsing,» INRIA.