Informe de las Prácticas Laborales: Gramáticas Matriciales Simples. Primera aproximación para una solución al problema SAT

José Jorge Rodríguez Salgado
Tutor: Msc. Fernando Raúl Rodríguez Flores

16 de julio de 2019

Resumen

Dentro de la amplia gama de formalismos creados para reconocer determinados lenguajes dependientes del contexto de manera eficiente se encuentran las gramáticas controladas matricialmente, y dentro de este conjunto las gramáticas matriciales simples. Estas últimas fueron definidas en [1] y, a la vez que adicionan algunas restricciones sobre sus producciones, poseen varias propiedades interesantes, algunas de las cuales se expondrán en el presente trabajo. En [2] y [3] se exponen soluciones a determinadas instancias del problema SAT usando gramáticas libres del contexto y de concatenación de rango respectivamente. En el presente trabajo se estudian las gramáticas matriciales simples y se valora la posibilidad de su uso para la solución eficiente de al menos un determinado conjunto de instancias del problema SAT.

1. Introducción

Desde el pasado siglo, se han definido varios formalismos que permiten reconocer, de manera eficiente, lenguajes dependientes del contexto. Estos formalismos han permitido grandes avances en campos como el procesamiento de lenguaje natural. Las gramáticas matriciales simples, que son un caso especial de las gramáticas controladas matricialmente, fueron definidas en [1], donde se demuestran numerosas propiedades de las mismas. Varias de estas propiedades sugieren que estos tipos de lenguajes pudieran resultar útiles en la solución del problema SAT.

El Problema de la Satisfacibilidad Booleana (SAT) recibe como entrada una fórmula de la lógica proposicional bivalente y su salida es si esta fórmula es satisfacible para al menos una interpretación de las variables que en ella ocurren. Sin perder generalidad se puede asumir que la fórmula de entrada viene dada en forma normal conjuntiva. Este fue el primer problema NP-completo descubierto aunque para algunas instancias, como el 2-SAT, es soluble en tiempo polinomial.

En [2] y en [3] se exponen métodos para solucionar otras instancias del SAT en tiempo polinomial utilizando gramáticas libres del contexto y de concatenación de rango respectivamente. En el presente trabajo se explora la posibilidad del uso de las gramáticas matriciales simples para solucionar al menos un conjunto de instancias del SAT de manera eficiente.

A continuación se han escrito tres secciones. En la sección 2 se muestran los contenidos principales que sobre los que se basa el resto del trabajo. En la sección número 3 se expone, la estrategia propuesta para hallar una solución a al menos un conjunto de instancias del SAT. Por último, en la sección 4 se realizan las conclusiones y las recomendaciones del presente trabajo.

En la siguiente sección se introducen algunos contenidos que resultan fundamentales para la presente investigación.

2. Preliminares

En la presente sección se brindan las definiciones y propiedades que conforman la base del presente trabajo. En el siguiente epígrafe se definen las gramáticas matriciales simples, las cuales son capaces de reconocer el lenguaje ω_m^n. O sea el lenguaje de las cadenas formadas por n copias de una misma cadena de longitud m. Este lenguaje permite, como se verá más adelante, reconocer todas las interpretaciones válidas de las variables que componen a un problema SAT. Este lenguaje puede ser intersectado con el de las interpretaciones que hacen verdadera la proposición de entrada del SAT si las variables fueran todas distintas y, de esta manera, se obtendría el lenguaje de las interpretaciones que hacen verdadera la proposición. En [2] y [3] se sigue esta estrategia, con la salvedad de que no se utilizan gramáticas matriciales simples y, por tanto, no se genera el lenguaje ω_m^n. En ambos se prosigue resolviendo el problema del vacío para el lenguaje intersección, de esta manera se sabe si la fórmula es satisfacible o no. En el presente trabajo se explora esta vía y una variante que no requiere resolver el problema del vacío.

2.1. Gramáticas Matriciales Simples

En este epígrafe se presentan las gramáticas matriciales simples. Se hace además un recorrido por las propiedades que resultan importantes para una posible solución del problema SAT. A continuación, la definición de una gramática matricial simple.

Definición 2.1 *Una gramática matricial simple de orden n (n-SMG) es un* $(n+3)$*-uplo* $G_n =< V_1, ..., V_n, S, P, \Sigma >$ *donde* $V_1, ..., V_n$ *son conjuntos disjuntos dos a dos de no terminales,* S *es el símbolo distinguido de la gramática,* $\forall i, 1 \leq i \leq n, S \notin V_i$. Σ *es el alfabeto del lenguaje y* P *es un conjunto de matrices de producciones que cumplan con una de las siguientes reglas:*

1. $[S \to \omega]$ *donde* $\omega \in \Sigma^*$
2. $[S \to a_{11}A_{11}a_{12}A_{12}...a_{1k}A_{1k}a_{21}A_{21}...a_{nk}A_{nk}b]$ *donde* $\forall i, j 1 \leq i \leq n$ *y* $1 \leq j \leq k, A_{ij} \in V_i$ *además* $a_{ij} \in \Sigma^*$ *y* $b \in \Sigma^*$
3. $[A_1 \to \omega_1, ..., A_n \to \omega_n]$ *donde* $\forall i, j 1 \leq i \leq n, A_i \in V_i$ *y* $\omega_i \in \Sigma^*$.
4. $[A_1 \to a_{11}A_{11}...a_{1k}A_{1k}b_1,...,A_n \to a_{n1}A_{n1}...a_{nk}A_{nk}b_n]$ *donde* $\forall i, j 1 \leq i \leq n y 1 \leq j \leq k$ *se tiene que* $A_i, A_{ij} \in V_i$, $a_{ij}, b_i \in \Sigma^*$

Definición 2.2 *Dada una n-SMG* $G_n =< V_1, ..., V_n, P, S, \Sigma >$*, se dice que a deriva directamente en b* ($a \Rightarrow b$) *con* $a, b \in (V_1 \cup ... \cup V_n \cup \{S\} \cup \Sigma^*)$ *si se cumple cualquiera de las siguientes condiciones:*

1. $a = S$ *y* $[S \to b] \in P$
2. *Existen* $y_1, ..., y_n \in \Sigma^*$ *y* $\forall i, 1 \leq i \leq n, \omega_i, z_i \in (V_i \cup \Sigma^*), A_i \in V_i$*;* $a = y_1A_1z_1...y_nA_nz_n$, $b = y_1\omega_1z_1...y_n\omega_nz_n$ *y* $[A_1 \to \omega_1, ..., A_n \to \omega_n] \in P$

En [1] se demuestra que el conjunto de los n-SML es cerrado bajo la intersección con un lenguaje regular. Además también se muestra cómo el problema del vacío para un n-SML se reduce al problema del vacío para un CFL en tiempo polinomial. Más adelante se verá cómo estas propiedades juegan un papel fundamental para la estrategia con la que se pretende resolver el SAT. En el siguiente epígrafe se define el problema SAT y se ahonda en algunas de sus propiedades.

2.2. Problema SAT

El problema de la satisfacibilidad booleana (SAT) recibe como entrada una fórmula de la lógica proposicional bivalente y devuelve como salida si esta es satisfacible para alguna interpretación de las variables que en ella intervienen.

Toda fórmula de la lógica proposicional bivalente puede escribirse en Forma Normal Conjuntiva (FNC) o sea, formada por una conjunción de disyunciones. Por lo tanto, en lo adelante se asume que la entrada del SAT viene dada en FNC.

Se denomina instancia de la variable x a toda ocurrencia de x en la fórmula de entrada. Se dice que la instancia de x tiene signo positivo si no está negada y signo negativo si lo está.

Las variables de una fórmula se pueden numerar y de esta forma establecer un orden entre las mismas, de forma tal que se dispongan de la forma $x_1, x_2, ..., x_m$ y se puede hacer referencia a la i-ésima variable con $1 \leq i \leq m$.

En todas las cláusulas (disyunciones) de una FNC no tienen que aparecer, en general, instancias de todas las variables. Por ejemplo en la fórmula $(x_1 v x_2) y (-x_1)$, la segunda variable no está instanciada en la segunda cláusula. Debe notarse, sin embargo, que toda fórmula en FNC puede reescribirse de forma tal que no aparezcan más de una instancia de la misma variable en la misma cláusula. Esto se debe a que si aparecen dos instancias de una misma variable con signo contrario en una misma cláusula, entonces esa cláusula siempre será verdadera. Por otra parte, $XvX = X$ donde X es una instancia de una variable ya sea con signo negativo o positivo.

A partir de lo expuesto en el párrafo anterior, se definirá a continuación una variante para escribir fórmulas en FNC utilizando solamente cuatro símbolos y haciendo que cada cláusula tenga la misma cantidad de estos símbolos.

Definición 2.3 *Dada una fórmula en FNC formada por instancias de n variables distintas numeradas de 1 a n. Se pueden reescribir sus cláusulas utilizando símbolos del conjunto $\{a, b, c\}$ de la manera siguiente:*

Para cada cláusula, se colocarán n símbolos. Si la variable i-ésima ($1 \leq i \leq n$) ocurre con signo positivo, se coloca una a en la posición i, si ocurre con signo negativo se coloca una b y si no ocurre se coloca una c. Se usa un caracter especial para separar las cláusulas, en este trabajo usaremos el caracter $|$. A esta forma de escribir las fórmulas en FNC se le llamará forma estándar.

Por ejemplo:

La fórmula usada anteriormente, $(x_1 v x_2) y (-x_1)$, escrita en notación estándar sería: $aa|bc$. Debido a que las dos variables aparecen en la primera cláusula con signo positivo y, en la segunda, cláusula la primera variable aparece con signo negativo mientras que la segunda no aparece.

La ventaja de utilizar la forma estándar para describir una fórmula proposicional en FNC es que todas las cláusulas están conformadas por la misma cantidad de caracteres. Esto permite que una cadena de la forma ω_m^n sobre el alfabeto $\{0, 1\}$ pueda verse como la asignación, a cada instancia de cada una de las m variables que ocurren en las n cláusulas de una fórmula en FNC, de un mismo valor de verdad. El lenguaje ω_m^n puede ser generado por una n-SMG, este se correspondería con el lenguaje de todas las posibles interpretaciones de las variables que ocurren en una fórmula de la lógica proposicional bivalente (asumiendo que $\omega \in \{0, 1\}^*$). Luego lo que se necesitaría sería desechar las interpretaciones que no satisfacen la fórmula, para ello se intersecta este lenguaje con el de las interpretaciones que satisfacen la fórmula, el cual, como se verá, es regular. Este proceso y el análisis de la satisfacibilidad de la fórmula a partir del lenguaje resultante se describe en detalle en la siguiente sección.

3. Estrategia de solución del SAT a partir de un n-SML y un lenguaje regular

En la presente sección se expone cómo, a partir del lenguaje ω_m^n y el lenguaje de las interpretaciones que satisfacen una fórmula (que si se escribe en FNC está conformada por m variables y n cláusulas), se puediera saber si la fórmula es o no satisfacible. Además, de ser satisfacible, se tendría una n-SMG que genera todas las interpretaciones que la satisfacen. Primeramente se muestra la n-SMG que genera ω_m^n.

3.1. El lenguaje ω_m^n

Este lenguaje se puede generar a partir de una n-SMG que contenga un total de $2 * m + 1$ matrices de producciones.

En efecto, sea $G_n =< \{A_1, A_{12}, ..., A_{1m}\}, \{A_2, A_{22}, ..., A_{2m}\}, ..., \{A_n, A_{n2}, ..., A_{nm}, P, S, 0, 1\} >$ donde el conjunto de matrices de producciones P contiene las siguientes matrices:

1. $[S \rightarrow A_1 A_2 ... A_n]$
2. $[A_1 \rightarrow 1A_{12}, ..., A_n \rightarrow 1A_n 2]$
3. $[A_1 \rightarrow 0A_{12}, ..., A_n \rightarrow 0A_n 2]$
4. Para $2 \leq i < m$, $[A_1 i \rightarrow 1A_{1(i+1)}, ..., A_n i \rightarrow 1A_n(i + 1)]$
5. Para $2 \leq i < m$, $[A_1 i \rightarrow 0A_{1(i+1)}, ..., A_n i \rightarrow 0A_n(i + 1)]$
6. $[A_1 m \rightarrow 1, ..., A_n m \rightarrow 1]$
7. $[A_1 m \rightarrow 0, ..., A_n m \rightarrow 0]$

Se puede demostrar fácilmente que el lenguaje generado por G_n $(L(G_n))$ es ω_m^n.

Como ya se ha planteado, este lenguaje puede verse como el de todas las posibles interpretaciones de una fórmula lógica en FNC y forma estándar. Ahora, para eliminar las interpretaciones que no satisfacen la fórmula se debe intersectar con el lenguaje de todas las asignaciones de valores a las instancias de las variables que satisfacen la fórmula. Los detalles a este respecto se ofrecen en el siguiente epígrafe.

3.2. El lenguaje de las interpretaciones que satisfacen una fórmula lógica

En [2] y [3] se construye el autómata finito que reconoce el lenguaje de las interpretaciones que satisfacen una fórmula si en esta no ocurren más de una instancia de una misma variable. De existir varias instancias de una misma variable este lenguaje contendría interpretaciones imposibles, o sea, que asignan a dos instancias distintas de una misma variable valores de verdad distintos. Por otra parte, el lenguaje ω_m^n, generado en el epígrafe anterior, es el de todas las interpretaciones válidas, las que asignan a cada instancia de una misma variable el mismo valor de verdad, pero contiene cadenas que no satisfacen la fórmula. Es por eso que ninguno de estos lenguajes por separado contienen exactamente a todas las interpretaciones válidas que satisfacen la fórmula. Sin embargo este último lenguaje es el resultado de la intersección de los anteriores.

En [1] se demuestra de manera constructiva que el conjunto de los n-SML es cerrado bajo la intersección con lenguajes regulares. Para hacerlo se define una n-SMG que genera este lenguaje. La misma contiene un total de $Q^2 N$ no terminales donde Q es la cantidad de estados del autómata finito y N es la cantidad de no terminales de la n-SMG inicial. La gramática que genera la intersección de ω_m^n con el lenguaje de todas las asignaciones que satisfacen la fórmula contiene, de manera general, una cantidad exponencial de producciones con respecto a la cantidad de variables que ocurren en la fórmula de entrada.

En este punto sería recomendable analizar bajo qué condiciones de la fórmula de entrada se pudiera reducir la cantidad de producciones para que esta estuviera acotada por una función polinomial de la cantidad de variables que ocurren.

Debido a la forma en la que se construye la gramática de la intersección se propone la siguiente conjetura, cuya demostración o refutación se pospone como trabajo futuro.

Una fórmula lógica proposicional bivalente es satisfacible si es posible construir la intersección de los lenguajes ω_m^n y de asignaciones que la satisfacen de la manera que se expone en [1]

No obstante es posible que en el futuro sea más factible construir alguna grámatica equivalente que sea n-SMG y entonces resolver el problema de si la misma genera un lenguaje vacío o no, de la misma forma que se hace en [2] y [3]. En [1] se demuestra que el problema del vacío para una n-SMG se reduce al problema del vacío de una CFG en tiempo polinomial. Este a su vez también se resuelve en tiempo polinomial.

Luego de analizar los posibles enfoques a seguir para lograr caracterizar al menos un conjunto de instancias del problema SAT que puedan ser solubles en tiempo polinomial utilizando lenguajes matriciales simples, se procede a realizar las conclusiones y recomendaciones del presente trabajo.

4. Conclusiones y recomendaciones

En este trabajo se ha propuesto una nueva vía para intentar caracterizar un conjunto de instancias del problema SAT utilizando las gramáticas matriciales simples. Para ello se definieron estas gramáticas y, específicamente el lenguaje ω_m^n. Se definió y trabajó con la forma estándar de una fórmula proposicional en FNC, a partir de la cual se puede tomar una cadena del lenguaje ω_m^n como una interpretación de las variables que ocurren en la fórmula. Tomando como referencia a [2] y [3] se siguió la línea de construir un autómata que reconociera las asignaciones que satisfacen la fórmula para luego intersectar este lenguaje con el de ω_m^n. Esta intersección es la que plantea un reto, pues, de manera, la gramática resultante contiene una cantidad exponencial de producciones con respecto a la cantidad de variables que contiene la fórmula de entrada. Finalmente se conjetura que el problema SAT es equivalente al de construir esta gramática intersección.

Es recomendable intentar demostrar o refutar la conjetura realizada así como investigar qué restricciones se le pudieran aplicar a las fórmulas de entrada para que el problema de decidir si es o no satisfacible sea soluble en tiempo polinomial.

Referencias

[1] *Oscar H. Ibarra. "Simple Matrix Languages". Department of Computer, Information and Control Sciences, University of Minneapolis, Minnesota 1970.*

[2] *Alina Fernández Arias. .El problema de la Stisfacibilidad Booleana Libre del Contexto. Un algoritmo polinomial". Universidad de La Habana. Junio 2007.*

[3] *Manuel Aguilera López. "Problema de la Satisfacibilidad Booleana de Concatenación de Rango Simple". Jornada Científica Estudiantil MATCOM 2016*

[4] *John E. Hopcroft y otros. Ïntroduction to Automata Theory, Languages and Computation, second edition". Addison-Wesley, 1942.*

1, 2, 4, 5

1, 2, 4, 5

1, 2, 4, 5