Chapter 1

Estado del arte

1.1 SAT

La versión más básica de SAT es el 2-SAT, en este problema cada cláusula tiene a lo sumo 2 variables, dicho problema se puede resolver mediante una modelación basada en grafos con un algoritmo polinomial. Pero si se aumenta la cantidad de variables el problema se vuelve mucho más complejo, para el caso del 3-SAT (cada cláusula tiene a lo sumo 3 variables) no se conoce algoritmo polinomial que permita resolverlo y está demostrado que cualquier instancia del SAT puede ser reducida a una instancia del 3-SAT.

En 1971, Stephen Cook demostró que SAT es NP-completo en su célebre artículo *The Complexity of Theorem-Proving Procedures*, estableciendo así el nacimiento de la teoría de NP-completitud. La demostración se basa en la construcción de una reducción polinómica desde cualquier problema en NP al 3-SAT, mostrando que todos los problemas de NP son, en esencia, tan difíciles como SAT.

Posteriormente, Leonid Levin, de manera independiente, llegó a conclusiones similares en el contexto de la Unión Soviética. Esta coincidencia histórica resalta la importancia de SAT en la computación teórica y su rol como punto de partida en el estudio de problemas intratables.

1.1.1 Problemas NP y la importancia de SAT

Después de la demostración de que SAT es NP-completo, surgió una cascada de descubrimientos de otros problemas NP-completos mediante reducciones polinómicas. Ejemplos emblemáticos incluyen el problema del *clique*, el del cubrimiento mínimo de vértices (*vertex cover*), el problema del viajante (*TSP*) y el problema de mochila (*backpack problem*), estos 3 últimos en su versión del problema de decisión.

La existencia de estas reducciones polinómicas refuerza la importancia de SAT como núcleo de la clase NP. Resolver SAT eficientemente tendría implicaciones revolucionarias para la informática, ya que permitiría resolver cualquier

problema en NP en tiempo polinómico.

1.1.2 Problemas SAT solubles en tiempo polinomial

Como se mencionó anteriormente no se conoce ningún algoritmo polinomial para resolver el problema SAT en general, pero existen casos particulares del problema que sí pueden ser resueltos en tiempo polinomial. A continuación se presentan los principales casos:

1. **1-SAT:** El problema 1-SAT es una instancia particular de SAT donde cada cláusula tiene a lo sumo un literal. Este problema puede ser resuelto en tiempo polinomial mediante un algoritmo de asignación de valores de verdad.

2. **2-SAT:** Como se mencionó anteriormente, el problema 2-SAT puede ser resuelto en tiempo polinomial mediante una modelación basada en grafos.

3. **Horn-SAT:** El problema Horn-SAT es una generalización del problema 2-SAT, donde cada cláusula tiene a lo sumo un literal positivo. Este problema puede ser resuelto en tiempo polinomial mediante el algoritmo de resolución de Horn.

1.2 Teoría de lenguajes

La teoría de lenguajes formales surgió como una rama de la computación teórica en la década de 1950, impulsada por los estudios de Noam Chomsky sobre gramáticas formales y su relación con los modelos de computación. En este contexto, Chomsky propuso en 1956 una jerarquía que clasifica los lenguajes formales en cuatro niveles de complejidad creciente, basándose en las restricciones impuestas a las gramáticas que los generan y a las máquinas que los reconocen.

Jerarquía de Chomsky

La jerarquía de Chomsky consta de los siguientes niveles:

1. **Lenguajes regulares:** Son los lenguajes más simples dentro de la jerarquía y son reconocidos por autómatas finitos. Se definen mediante gramáticas de tipo 3, en las cuales las reglas de producción tienen una forma estrictamente restringida, como $A \to aB$ o $A \to a$, donde A y B son no terminales y a es un símbolo terminal. Estos lenguajes tienen aplicaciones prácticas en el análisis léxico y el diseño de patrones.

2. **Lenguajes libres de contexto *CFL*:** Reconocidos por autómatas con pila, los lenguajes libres de contexto son definidos por gramáticas de tipo 2. En estas gramáticas, las reglas de producción tienen la forma $A \to \alpha$, donde α es una cadena de terminales y no terminales. Los lenguajes CFL

son fundamentales para analizar la sintaxis de los lenguajes de programación y para modelar estructuras jerárquicas.

3. **Lenguajes sensibles al contexto:** Estos lenguajes son más expresivos y están definidos por gramáticas de tipo 1, que permiten reglas de producción de la forma $\alpha A\beta \rightarrow \alpha\gamma\beta$, donde A es un no terminal, α y β son cadenas de símbolos, y γ no es una cadena vacía. Los lenguajes sensibles al contexto son reconocidos por máquinas de Turing linealmente acotadas.

4. **Lenguajes recursivamente enumerables:** Estos lenguajes corresponden a los más generales y son definidos por gramáticas de tipo 0, que no tienen restricciones en las reglas de producción. Son reconocidos por máquinas de Turing y representan la clase completa de problemas que pueden ser computados.

El **problema de la palabra** es una cuestión fundamental en la teoría de lenguajes y las ciencias de la computación. Formulado inicialmente en el contexto de la teoría de grupos, el problema plantea:

> Dada una representación finita de un lenguaje formal y una cadena de símbolos, ¿es posible determinar si esta cadena pertenece al lenguaje?

En términos de computación, el problema de la palabra está estrechamente relacionado con la decidibilidad y la complejidad computacional. La solución a este problema depende del modelo computacional que se utilice para definir el lenguaje.

Todo problema en Ciencias de la Computación puede ser reducido a un problema de la palabra, ya que cualquier problema puede ser codificado como un lenguaje formal. Por lo tanto, el estudio del problema de la palabra es fundamental para comprender la computabilidad y la complejidad de los problemas computacionales. De ahí que en el presente trabajo se le preste especial atención a este problema enfocado en el contexto del SAT.

1.3 SAT y la teoría de lenguajes

Como parte del estudió del problema SAT, se ha desarrollado una línea de investigación en la facultad utilizando un enfoque novedoso basado en formalismos de la teoría de lenguajes, buscando resolver instancias específicas del problema.

1.3.1 Problema satisfacibilidad booleana libre del contexto

El primer trabajo desarrollado como parte de esta línea de investigación [1] consiste en resolver el problema satisfacibilidad booleana libre del contexto (*CF-SAT*), el cual es una instancia específica del SAT donde las variables donde la fórmula booleana es una fórmula booleana libre del contexto.

Una fórmula booleana se considera libre del contexto si para cualquier par de instancias de una variable x_i y x_j con $i < j$ se cumple que si existe otra variable con instancia x_k con $i < k < j$ entonces todas las instancias de esta nueva variable ocurren entre x_i y x_j.

El CF-SAT consiste en transformar una fórmula booleana en una lista de instancias de variables, donde se asume que 2 instancias de una variable no tienen por qué tener el mismo valor de verdad. Luego se define un autómata que dada una cadena de 0 y 1 y una fórmula booleana, determina si se obtiene un valor de verdad para la fórmula booleana donde cada instancia de una variable toma el valor de verdad que se corresponde con la cadena de 0 y 1, dicho autómata se denomina autómata booleano.

Entonces para verificar que 2 instancias de una variable tengan el mismo valor de verdad se intersecta dicho autómata con una gramática libre del contexto *CFG* obteniendo un autómata de pila (esto es posible por la estructura de la fórmula booleana libre del contexto planteada anteriormente). Después de esto se plantea un algoritmo para dado este autómata de pila, generar todas las posibles cadenas de 0 y 1 que se corresponden con una asignación de valores de verdad y como consecuencia se obtiene un generador de todas las posibles asignaciones de valores de verdad para una fórmula booleana. Luego para determinar si la fórmula es satisfacible solo queda verificar si este conjunto de soluciones es no vacío.

1.3.2 Problema de la satisfacibilidad para gramáticas de concatenación de rango simple

El próximo trabajo relacionado con este tema realizado en la facultad [2] consistió en definir y analizar el problema de la satisfacibilidad para gramáticas de concatenación de rango simple. En este trabajo se toma el mismo enfoque que el anterior, pero en vez de intersectar el autómata booleano con una gramática libre del contexto se intersecta con una gramática de concatenación de rango simple dando lugar a la resolución de un conjunto más amplio de problemas SAT que el problema anterior.

1.3.3 Problema de la satisfacibilidad para gramáticas matriciales

Continuando la línea del autómata booleano empleado en la resolución de instancias del SAT el próximo trabajo desarrollado [3] consistió en analizar las gramáticas matriciales. Nuevamente intersectando el autómata booleano con un formalismo que cuente con un algoritmo para comprobar el problema del vacío en tiempo polinomial en este caso se eligieron las gramáticas matriciales que nuevamente ofrecen un conjunto más amplio de problemas que el CF-SAT.