

Una aproximación al lenguaje de todas las fórmulas booleanas satisfacibles

Raudel Alejandro Gómez Molina

Facultad de Matemática y Computación
Universidad de La Habana

10 de febrero de 2025

Tutor: MSc. Fernando Raul Rodriguez Flores

Objetivos del trabajo

- ▶ Objetivo general: Definir y construir el lenguaje de todas las fórmulas booleanas satisfacibles
- ▶ Objetivos específicos:
 - ▶ Estudiar el estado del arte
 - ▶ Codificación de una fórmula booleana en una cadena
 - ▶ Definir el lenguaje de todas las fórmulas booleanas satisfacibles L_{S-SAT}
 - ▶ Construir L_{S-SAT} utilizando una transducción finita
 - ▶ Construir L_{S-SAT} utilizando gramáticas de concatenación de rango y sin usar la transducción finita.
 - ▶ Analizar las implicaciones computacionales

Gramáticas de concatenación de rango (RCG)

- ▶ Un rango es un intervalo de la cadena
- ▶ A los no terminales se les llama predicados
- ▶ Cada predicado tiene una secuencia de argumentos y recibe un vector de cadenas
- ▶ Cada argumento está formado por variables y no terminales

$$A(aX, cbYZ)$$

- ▶ Una sustitución de rango asocia una variable a un rango
- ▶ A las producciones se les llama cláusulas

$$A(x_1, \dots, x_k) \rightarrow B_1(y_{1,1}, \dots, y_{1,m_1}) \dots B_n(y_{n,1}, \dots, y_{n,m_n}),$$

Gramáticas de concatenación de rango (RCG)

- ▶ Cuando se deriva en las RCG
 - ▶ Se asocia cada cadena del vector que recibe el predicado izquierdo a un argumento
 - ▶ Se hace la sustitución de rango para cada argumento obteniendo los valores de las variables
 - ▶ Con los valores de las variables se construyen los vectores con los que se evalúa en los predicados derechos
 - ▶ Se repite el mismo proceso en cada uno de los predicados derechos
- ▶ Un predicado reconoce un vector si existe una secuencia de derivaciones y sustituciones de rango, tales que se deriva en la cadena vacía

Codificación de una fórmula booleana en una cadena

- ▶ a : la variable está sin negar en la cláusula
- ▶ b : la variable está negada en la cláusula
- ▶ c : la variable no está en la cláusula
- ▶ d : separador para delimitar una cláusula

$$(x_1) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_2 \vee x_3) \Leftrightarrow acc\mathbf{d}ab\mathbf{a}d\mathbf{c}b\mathbf{a}d$$

- ▶ $L_{FULL-SAT}$ lenguaje de todas las fórmulas booleanas en forma normal conjuntiva
- ▶ L_{SAT} lenguaje de todas las fórmulas booleanas satisfacibles

Asignación de variables para una fórmula booleana

- ▶ $L_{0,1,d} = \{(wd)^+ \mid w \in \{0,1\}^+\}$
- ▶ $r \in L_{0,1,d}$ representa una asignación de valores y $e \in L_{FULL-SAT}$ representa una fórmula booleana: se debe cumplir que $|e| = |r|$ y que e tenga la misma cantidad de caracteres d que r
- ▶ $r = 101\mathbf{d}101\mathbf{d}101\mathbf{d}$ y $e = abcd\mathbf{c}bb\mathbf{d}accd$,

$$(x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1)$$

$$(true \vee \neg false) \wedge (\neg false \vee \neg true) \wedge (true) = true$$

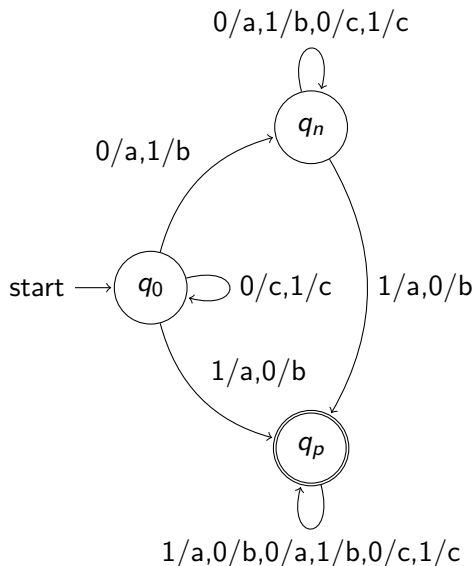
Transductor T_{CLAUSE}

Entrada: $r \in L_{0,1,d}$

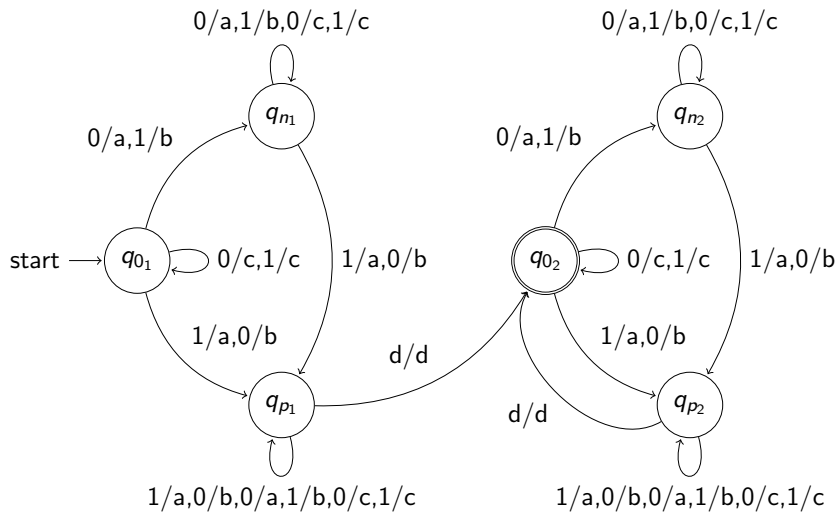
Salida: $e \in L_{FULL-SAT}$

Estados de T_{CLAUSE}

- ▶ q_0 : estado inicial
- ▶ q_p : representa el estado positivo (estado de aceptación)
- ▶ q_n : representa el estado negativo



Transducer T_{SAT}



$L_{0,1,d}$ como lenguaje de concatenación de rango

Producciones de la gramática $G_{0,1,d}$

- ▶ $S(X) \rightarrow A(X)$
 - ▶ $A(XdY) \rightarrow B(Y, X)C(X)$
 - ▶ $B(XdY, P) \rightarrow B(Y, P)C(X)Eq(X, P)$
 - ▶ $B(\varepsilon, P) \rightarrow \varepsilon$
-
- ▶ C comprueba que la cadena está formada por 0 y 1
 - ▶ Eq comprueba que 2 cadenas sean iguales

Construcción de L_{SAT} mediante una RCG

Las producciones de la gramática G_{S-SAT} se agrupan en 4 grupos (fases):

1. Derivación inicial de la gramática
2. Mientras se reconoce la cadena se generan todas las posibles interpretaciones que hacen verdadera la primera cláusula
3. Comprobar que la interpretación generada en la primera fase satisface el resto de las cláusulas
4. Comprobar si la interpretación generada en la primera fase satisface una cláusula

Producciones de G_{S-SAT}

Producciones agrupadas por fases

1. $S(X) \rightarrow A(X)$
2. Compuesta por los predicados A , P (estado positivo) y N (estado negativo)

$$H(_X, Y) \rightarrow J(X, Y_)$$

3. $B(X_1 d X_2, Y) \rightarrow C(X_1, Y)B(X_2, Y)$ y $B(\varepsilon, Y) \rightarrow \varepsilon$
4. Compuesta por los predicados C , Cp (estado positivo) y Cn estado negativo

$$H(_X, _Y) \rightarrow J(X, Y)$$

Resultados derivados de T_{SAT} y G_{S-SAT}

Resultados derivados de T_{SAT}

- ▶ El problema de la palabra para todo formalismo que genere $L_{0,1,d}$ y sea cerrado bajo transducción finita, es NP-Duro
- ▶ Se conjetura que todo formalismo que genere $L_{0,1,d}$ tiene tamaño $O(1)$ en su representación

Resultados derivados de G_{S-SAT}

- ▶ No es necesaria la transducción finita de T_{SAT} para construir L_{S-SAT}
- ▶ Las RCG reconocen todos los problemas de la clase NP-Completo en su representación como lenguaje formal

Problemas propuestos

Instancias polinomiales de SAT

- ▶ Encontrar una RCG que permita reconocer el 2-SAT y cuyo problema de la palabra sea polinomial

Nuevo formalismo basado en las RCG

- ▶ Investigar qué propiedades de las RCG limitan que estas no sean cerradas bajo transducción finita
- ▶ Construir un nuevo formalismo basado en las RCG que sea cerrado bajo transducción finita
- ▶ Comprobar si este formalismo es capaz de describir el lenguaje $L_{0,1,d}$

Una aproximación al lenguaje de todas las fórmulas booleanas satisfacibles

Raudel Alejandro Gómez Molina

Facultad de Matemática y Computación
Universidad de La Habana

10 de febrero de 2025

Tutor: MSc. Fernando Raul Rodriguez Flores

Muchas gracias