Lógica Computacional 2019-2

Proyecto 1

Atacando el problema SAT: Métodos sintácticos vs semánticos

Lógica Computacional 2019-2
Profesor: Fernando Abigail Galicia Mendoza
Ayudante: Leonardo Hernández Cano
Laboratorista: Francisco Emmanuel Anaya Gónzalez
Licenciatura en Ciencias de la Computación
Facultad de Ciencias, UNAM

Hernández Leyva Mirén Jessamyn
Ruiz López Jorge Antonio

1. Introducción.

Para comenzar con esto es necesario definir varios conceptros previos que nos servirán de apoyo. Uno de los conceptos más importantes por definir es la *Lógica matemática*. Esta es una rama de las matemáticas que se encarga de estudiar los razonamientos y demostraciones, además de proporcionar las herramientas para ser capaces de inferir una "conclusión" correcta a partir de unas afirmaciones o proposiciones previas que denotaremos como "premisas".

La lógica matemática estudia la inferencia mediante la construcción de sistemas formales como la lógica proposicional, la lógica de primer orden o la lógica modal. Estos sistemas se encargan de capturar las características escenciales de las inferencias válidas en los lenguajes naturales, pero al ser estructuras formales de análisis matemático, permiten realizar demostraciones rigurosas sobre ellas, y así mismo poder afirmar o negar un razonamiento de forma matemática.

Este tipo de razonamiento en ciencias de la computación es de vital importancia ya que nos permite verificar la correctez de ciertos algoritmos que podamos implementar, además de utilizarse en la programación lógica y en el análisis y optimización (de recursos temporales y espaciales) de algoritmos, lo cual busca reducir costos en tiempo y espacio al momento de ejecutarlos.

En Ciencias de la Computación nos interesa saber la satisfacibilidad de una fórmula (φ) porque, podemos en ciertas ocasiones ver un problema como una secuencia de pasos (premisas-conclusión) lo cual sería un fórmula. Saber que una fórmula es satisfacible significa que podemos dar una interpretación (I) para dicha fórmula en la cual $I(\varphi) = 1$ para cualquier φ.

2. Lógica Proposicional.

El lenguaje español puede presentar disntitas ambigüedades dependiente del lector, el recpetor y de la interpretación que le de cada uno. Cierta parte del español podemos analizarla mediante fórmulas lógicas y así dar un analisis más profundo al "texto" que se recibe y de esta forma evitar aquelas posibles ambigüedades que puedan presentarse. Para realizar dicho análisis podemos apoyarnos de:

- Argumentos lógicos:
  Es una colección finita de afirmaciones $A_1 ... A_n$ (proposiciones) que se divide en premisas y conclusión. Las premisas y conclusión tienen que poder recibir un valor de verdad y el argumento puede ser correcto o incorrecto.

- Proposiciones:
  Enunciados a los que se les puede asignar un valor de verdad (1,0) dependiendo de la interpretación que se le asigne.

- Conectivos:
  Son operadores lógicos entre las proposiciones.

  - Negación: Se refiere a las expresiones que llevan un No, como pueden ser: no es cierto que, es falso que, etc. Se denota con el símbolo $\neg$ y su tabla de verdad es:

P	$\neg$ P
1	0
0	1

  - Conjunción: Corresponde con el "y" o "pero". Se denota con el simbolo $\wedge$ y su tabla de verdad es:

P	Q	P $\wedge$ Q
1	1	1
1	0	0
0	1	0
0	0	0

- Disyunción: Corresponde con la o. Se denota con el simbolo $\vee$ y su tabla de verdad es:

P	Q	P $\vee$ Q
1	1	1
1	0	1
0	1	1
0	0	0

- Implicación: Corresponde con el si, entonces, sólo si, es condición suficiente. Se denota con el simbolo $\rightarrow$ y su tabla de verdad es:

P	Q	P $\rightarrow$ Q
1	1	1
1	0	0
0	1	1
0	0	1

- Equivalencia: Corresponde con el es quivalente, si sólo si, es condición necesaria y suficiente. Se denota con el simbolo $\longleftrightarrow$ y su tabla de verdad es:

P	Q	P $\longleftrightarrow$ Q
1	1	1
1	0	0
0	1	0
0	0	1

- Premisas: Cada una de las proposiciones de un argumento lógico anteriores a la conclusión.

- Conclusión: Es una proposición al final de un argumento lógico, luego de las premisas. Si el argumento es valido, entonces las premisas implicaran una conclusión.

- Tautologias: Aquellas fórmulas que se evalúan a verdadero en todos los estados posibles.

- Contradicciones: Aquellas fórmulas que se evalúan a falso en todos los posibles estados.

- Argumentos Correctos: El argumento $A_1, A_2..., A_n/\therefore B$ es correcto si y solo si $\models A_1, A_2..., A_n \rightarrow$ B

Ejemplo:

Los libros son baratos o son caros.	Premisa 1
Los libros no son caros.	Premisa 2
.	
Los libros son baratos	Conclusión

Para determinar la veracidad de una fórmula podemos basarnos en el principio de refutación. (demostrar)

Interpretación: Asignación de un valor de verdad a las variables que constituyen las fórmulas.

Ejemplo 1:

"Si Pepito compite en natación va a ganar el primer lugar. Si Juanito compite en natación va a ganar el primer lugar. Alguno de los dos no va a quedar en primer lugar en la competencia de natación. Por lo tanto o Pepito no compite o Juanito no copite."
Esto se puede traducir a la logica formal de la siguiente manera:
p: Pepito compite en natación.
q: Pepito gana.
r: Juanito compite en natación.
s: Juanito gana.

$p \rightarrow q$
$r \rightarrow s$

$\neg q$ v $\neg s$

$\neg p$ v $\neg r$

EJEMPLO 2:
"Si hoy es viernes, iré al cine. Hoy no es viernes por lo tanto no ire al cine."

Esto se puede traducir a la logica formal de la siguiente manera:
p : Hoy es viernes.
q : Ire al cine.

$p \rightarrow q$
$\neg p$

$\neg q$

EJEMPLO 3:
"Voy voy a comer tacos o una torta, si hoy como torta mi mamá comera quesadillas, si como tacos mi mamá come pambasos. Mi mamá comio pambasos por lo tanto yo comi tacos."

Esto se puede traducir a la logica formal de la siguiente manera:
p : Hoy comere tacos.
q : Hoy comere torta.
r : Mi mamá comera quesadillas.
s : Mi mamá comera pambasos.

$p \rightarrow q$
$q \rightarrow r$
$p \rightarrow s$
s

---

p

Algo que podría interesarnos es hacer un razonamiento de tipo ecuacional sobre las fórmulas, un ejemplo puede ser el siguiente, donde buscamos acortar una fórmula:

$$r \vee (p \longleftrightarrow (q \longleftrightarrow \neg q)) \equiv \perp$$

Pero sabemos que:

$$q \longleftrightarrow \neg q \equiv \perp$$

entonces

$$r \vee (p \longleftrightarrow (q \longleftrightarrow \neg q)) \equiv r \vee (p \longleftrightarrow \perp)$$

$$I(q \longleftrightarrow \neg q) = 0 = I(\perp)$$

Y de esto podemos obtener:

$$I(q) = I(\neg q)$$

y es una contradicción.

Realizar un procedimiento como el anterior es un razonamiento ecuacional sobre las formulas.
Además contamos con propiedades como siguen:

- Reflexividad

$$\mathrm{X} = \mathrm{X}$$

- Conmutatividad

$$\mathrm{X} = \mathrm{Y} \rightarrow \mathrm{Y} = \mathrm{X}$$

- Transitividad

$$\text{Si X} = \text{Y} \wedge \text{Y} = \text{Z} \rightarrow \text{X} = \text{Z}$$

- Regla de Leibniz

$$\text{Si X} = \text{Y entonces E[z:=X]} = \text{E [z:=Y]}$$

- Interpretaciones: Dado un estado de las variables $\mathcal{I}$: VarP $\rightarrow$ *Bool*, definimos la interpretación de las formulas respecto a $\mathcal{I}$ como la función $\mathcal{I}$: PROP $\rightarrow$ *Bool* tal que:

  - $\mathcal{I}(\top) = 1$
  - $\mathcal{I}(\bot) = 0$
  - $\mathcal{I}(\neg\varphi) = 1$ si y solo si $\mathcal{I}(\varphi) = 1$
  - $\mathcal{I}(\varphi \wedge \psi) = 1$ si y solo si $\mathcal{I}(\varphi) = \mathcal{I}(\psi) = 1$
  - $\mathcal{I}(\varphi \vee \psi) = 0$ si y solo si $\mathcal{I}(\varphi) = \mathcal{I}(\psi) = 0$
  - $\mathcal{I}(\varphi \rightarrow \psi) = 0$ si y solo si $\mathcal{I}(\varphi) = 1$ e $\mathcal{I}(\psi) = 0$
  - $\mathcal{I}(\varphi \leftrightarrow \psi) = 1$ si y solo si $\mathcal{I}(\varphi) = \mathcal{I}(\psi)$

Después de todo lo anterior podemos preguntarnos, ¿Cómo sabemos la satisfacibilidad de una fórmula? Pues podemos realizar un cierto procedimiento con el principio de refutación que nos ayude a saberlo.

Definición: Sea $\Gamma = \{A_1, ..., A_n\}$ un conjunto de fórmulas y B una formula. Decimos que B es consecuencia logica de Γ si toda interpretación $\mathcal{I}$ que satisface a Γ tambien satisface a B. Es decir si todo modelo de Γ es modelo de B.
Notacion: $\Gamma \models \mathcal{I}$

Ejemplo:

$$\Gamma = \{\ \neg q \rightarrow \neg r\ ,\ \neg r \rightarrow \neg p\ ,\ \neg p \rightarrow \neg q\} \models q \rightarrow r$$

Sea $\mathcal{I}$ un modelo de Γ. Tenemos que demostrar que $\mathcal{I}(q \rightarrow r) = 1$. Si $\mathcal{I}(q) = 0$ entonces $\mathcal{I}(q \rightarrow r) = 1$ y terminamos. En otro caso $\mathcal{I}(q) = 1$ donde $\mathcal{I}(q \rightarrow r) = 1$

Nos basamos en el teorema de la deducción para generar el ejemplo anteriorel cual se enuncia como sigue:
Sea Γ un conjunto de propocisiones y φ, $\psi \in Prop$
$\Gamma \cup \varphi \models \psi \leftrightarrow \Gamma \models \varphi \to \psi$
Se puede demostrar que este es valido de la siguiente manera:
Por hipotesis sabemos que $\Gamma \cup \varphi \models \psi$ entonces $\mathcal{I}(\Gamma \cup \varphi) = 1$ entonces $\mathcal{I}(\psi) = 1$ y por lo anterior $\mathcal{I}(\Gamma) = 1$ y la $\mathcal{I}(\varphi) = 1$, entonces como la $\mathcal{I}(\varphi) = 1 = \mathcal{I}(\psi)$, así se cumple que $\mathcal{I}(\varphi \to \psi) = 1$.
Ahora tomamos como hipotesis que $\Gamma \models \varphi \to \psi$
Si $\Gamma \models \varphi \to \psi$ tenemos que $\mathcal{I}(\Gamma \cup \varphi) = 1$ y por esto sabemos que $\mathcal{I}(\Gamma) = 1$ y $\mathcal{I}(\varphi) = 1$ pero por hipotesis sabemos que la $\mathcal{I}(\varphi \to \psi) = 1$ y como $\mathcal{I}(\varphi) = 1$ entonces $\mathcal{I}(\psi) = 1$, entonces $\Gamma \cup \varphi \models \psi$

Buscamos dar una "forma" a las fórmulas ya que esto reduce las llamadas y el costo a nivel computacional.

- Equivalencia: Decimos que dos expresiones son equivalentes si y sólo si en todos sus posibles estados se evalúan a lo mismo.
- Equivalencia lógica: sean dos formulas A y B. Si $A \leftrightarrow B$ es una tautología entonces decimos que A y B son lógicamente equivalentes. Denotado: $A \equiv B$

Dicho lo anterior podemos dar una forma a una fórmula como sigue:

- Forma Normal Negativa

  Definición:
  φ esta en FNN si y solo si

  - φ no contiene equivalencias ni implicaciones.
  - Las negaciones de φ solo afectan a formulas atomicas
  - Son todas.

  Algoritmo:

  - Eliminar todas las implicaciones y las equivalencias
  - Si tenemos una negación afectando a una PROP entonces distribuimos hasta que las negaciones solo estén en ATOM
    Ej:
    $$\neg(p \wedge q) \equiv (\neg p \vee \neg q)$$

- Forma Normal Conjuntiva

  Definición:
  φ esta en FNC si y solo si si es una conjunción de disyunciones de literales. Es decir, es de la forma
  $\varphi = (L_{1,1} \vee ... \vee L_{(1,n_1)}) \wedge ... \wedge (L_{m,1} \vee ... \vee L_{m,n_m})$.
  Algoritmo:

  - Obtener la FNN

- Distribuir
- Opcionalmente, reducir como sigue:

$$\perp \vee \varphi \equiv \varphi$$

y

$$\varphi \vee \varphi \equiv \varphi$$

Finalmente podemos definir el algortimo de manera más formal para poder encontrar la FNN de cualquier fórmula como se sigue:

Definiendo FNN:

- Si F $\in ATOM, entonces$F y $\neg F$ son Formas Normales Negativas (FNN).

- Si F y G son FNN entonces $(F \wedge G)$ y $(F \vee G)$ también lo son.

Ahora definiremos un algortimo sencillo para encontrar a partir de una fórmula F una fórmula G; su FNN, tal que $F \equiv G$.

- Eliminación de equivalencias.
  p $\longleftrightarrow q \equiv (p \rightarrow q)\wedge$ (q $\rightarrow p)$

  Eliminación de implicaciones.
  $p \rightarrow q \equiv \neg p\vee$ q

- Interiorización de negaciones.
  $\neg(p \wedge q) \equiv \neg p \vee \neg q$
  $\neg(p \vee q) \equiv \neg p \wedge \neg q$
  $\neg\neg p \equiv p$
  $\neg(\neg(p \vee q)) \equiv p\vee$ q
  $\neg((\neg p) \vee q) \equiv p \wedge \neg q$

3. Forma Normal Conjuntiva.

La forma normal conjuntiva (FNC) es la representación de cualquier formula como una conjunción de disyunciones $(C_1 \wedge C_2 \wedge C_3 \wedge ... \wedge C_n)$. Se puede demostrar que cualquier formula se puede representar en FNC con los siguientes pasos:

- Obtener la FNN
- Mediante las leyes distributivas se reescriben las conjunciones y disyunciones

$$(p \wedge q) \vee (p \wedge r) \equiv p \wedge (q \vee r)$$
$$(p \vee q) \wedge (p \vee r) \equiv p \wedge (q \wedge r)$$

Una buena razón para usar la FNC de una formula es el hecho de que simplifica el procedimiento para decidir si una fórmula es una tautología. Ya que Una cláusula $C = l_1 \vee l_2 \vee ... \vee l_n$ es tautología $(\Gamma = \Phi)$ si y sólo si existen 1 $\leq$ i, j $\leq$ n tales que $l_i^c = l_j$ Es decir, $\models C$ si y solo si para cada clausula C tiene a su literal complementaria.

Ejemplo:

$$\Gamma = \{p \rightarrow r \wedge r \rightarrow s \wedge \neg s \vee p\}$$

Queremos ver que $\mathcal{I}(\Gamma = 1)$, lo que es igual a que $\mathcal{I}(p \rightarrow r) = 1$, $\mathcal{I}(r \rightarrow s) = 1$, $\mathcal{I}(\neg s) = 1$ y $\mathcal{I}(\neg p) = 1$. Entonces inferimos lo que sigue por las reglas para interpretaciones:

1. $\mathcal{I}(\neg s) = 1$ si y solo si $\mathcal{I}(s) = 0$
2. $\mathcal{I}(r \rightarrow s) = 1$ si y solo si $\mathcal{I}(r) = 0$ porque $\mathcal{I}(s) = 0$
3. $\mathcal{I}(p \rightarrow r) = 1$ si y solo si $\mathcal{I}(p) = 0$ porque $\mathcal{I}(r) = 0$

Ahora si convertimos Γ primero en FNC es equivalente a:

$$\neg p \vee r \wedge \neg r \vee s \wedge \neg s \vee p$$

Y vamos eliminando paso a paso las literales que tengan a su complemento.

1. $r \wedge \neg r \vee s \wedge \neg s$ eliminando p y p^c
2. $s \wedge \neg s$ eliminando r y r^c
3.

Cada clausula tiene su complementaria por lo tanto es tautologia y por tanto es satisfacible.
Utilizamos la resolución binaria para decidir la correctitud de un argumento logico y se define como sigue:

$$\frac{C_1 \vee \ell \qquad\qquad \ell^c \vee C_2}{C_1 \vee C_2}$$

Cuando se llega a una clausula y su complementaria se devuelve la clausula vacia

$$\frac{p \qquad \neg p}{\square}$$

La FNC esta altamente ligada con el problema SAT ya que el problema SAT dado un conjunto de $varP$ y un conjunto de clausulas C se busca dar un modelo donde las interpretaciones satisfagan a C. Como vimos la forma normal conjuntiva nos permite definir una función que nos diga si una formula es una tautologia por lo tanto nos permite saber si el conjunto $varP$ y el conjunto de clausulas C es satisfacible.

Además con la resolución binaria podemos definir el algoritmo de saturación a partir de ella.
Primero definimos los metodos de saturación por medio de la resolución :

$$\mathcal{R}es_0(\mathbb{S}) = \mathbb{S}$$
$$\mathcal{R}es_{n+1}(\mathbb{S}) = \mathcal{R}(\mathcal{R}esn(\mathbb{S}))$$

$\mathbb{S}$ es un conjunto finito de cláusulas, $\mathbb{S}$ es no satisfacible si y sólo si $\square \in \mathcal{R}es_n(\mathbb{S})$ para alguna $n \in N$, apartir de ello creamos un método para verificar si un conjunto de cláusulas $\mathbb{S}$ es insatisfacible, construyendo los conjuntos $\mathcal{R}$esn($\mathbb{S}$) hasta hallar $\square$. En el caso que $\square$ nunca se genere en algún momento se tiene $\mathcal{R}$esn($\mathbb{S}$) = $\mathcal{R}\ es_{n+1}$ ($\mathbb{S}$) por lo que no hay más resolventes posibles, pero $\Box \notin \mathcal{R}\text{esn}(\mathbb{S})$. En este caso es satisfacible. Y por último si la saturación nunca termina pero tampoco genera $\Box$ se tiene la certeza de que para cualquier n $\in$ N, $\Box \in \mathcal{R}\text{esn}(\mathbb{S})$, en este caso no podemos decidir si es satisfacible o no.

Ejemplos:

– $\{p \to q, \neg(q \vee s), s \vee q\}$. Convertimos a FNC $\{\neg p \vee q, \neg q, \neg s, s \vee q\}$

1. $\neg p \vee q$ $\quad$ $Hipotesis.$
2. $\neg q$ $\quad$ $Hipotesis.$
3. $\neg s$ $\quad$ $Hipotesis.$
4. $s \vee q$ $\quad$ $Hipotesis.$
5. $\neg$ $\quad$ $Res(2, 4)$
6. $\Box$

llegamos a la clausula vacia por lo tanto es insatisfasible.

– $\{p \to r, r \to s, \neg s\}$. Convertimos a FNC $\{\neg p \vee r, \neg r \vee s, \neg s\}$

1. $\neg p \vee r$ $\quad$ $Hipotesis.$
2. $\neg r \vee s$ $\quad$ $Hipotesis.$
3. $\neg s$ $\quad$ $Hipotesis.$
4. $\neg r$ $\quad$ $Res(2, 3)$
5. $\neg p$ $\quad$ $Res(1, 4)$

como terminamos de desarrollar la formula y nunca se genera la clausula vacia entonces es satisfasible.

– $\{\neg p \vee \neg p, p \vee p\}$

1. $\neg p \vee \neg p$ $\quad$ $Hipotesis.$
2. $p \vee p$ $\quad$ $Hipotesis.$
3. $p \vee \neg p$ $\quad$ $Res(1, 2)$
4. $\neg p \vee p$ $\quad$ $Res(1, 2)$

5. $p \vee q$ $\qquad$ $Res(3,4)$

Como podemos notar se cicla pues el paso 5 es igual al paso 2, lo cual quiere decir que no se puede determinar su satisfacibilidad.

4. Tableux.

El tableux es un arbol cuya función es buscar una interpretación para determinada fórmula, sin embargo tambien es una herramienta muy útil cuando de satisfacibilidad se trata ya que con este podemos no solo determinar si una formula es o no satisfacible, tambien podemos decir si es una tautologia e inclusive un modelos para la fórmula.

Veamos ahora sus reglas de construcción:

1. La fórmula para la que deseamos construir el tableux aparece como raíz del árbol (esta debe consistir unicamente en conjunciones disyunciones o formulas atomicas).

2. Si el esquema de la fórmula es uan disyunción $(A \vee B)$, de la raiz del subárbol se abren dos ramás, una para la fórmula A y otra para la fórmula B (Recordando que el operador $\vee$ es conmutativo y asociativo podemos intercambiar el orden de las ramas y abrir la disyunción que queramos primero).

   (1) $p \vee q$

   (2) p q

3. Si el esquema de la fórmula es una conjuncion $(A \wedge B)$ entonces se pone a uno de lso operandos como hijo del otro (Como el operador $\wedge$ es conmutativo no importa cual elijamos como padre y cual cómo hijo).

   (1) $p \wedge q$
   (2) p
   (3) q

4. Aplicar la regla 2 y 3 hasta terminar de desglosar cada rama, donde las hojas sean solo formulas atomicas o negaciones de literales y contantes, o la rama se cierre (para ahorras trabajo).

Algunos ejemplos:

Ejemplos 1:

(1) $(p \vee \neg q) \wedge (r \vee \neg q)$

(2) $p \vee \neg q$

(3) $r \vee \neg q$

(4) p $\qquad\qquad$ $\neg q$

(5) $r \vee \neg q$ $\qquad\qquad$ $r \vee \neg q$

(6) r $\qquad$ $\neg q$ $\qquad$ r $\qquad$ $\neg q$

Ejemplos 2:

(1) $\neg(p \vee q) \vee (p \leftarrow q)$

(2) $\neg(p \vee q)$ $\qquad\qquad$ $p \to q$

Ejemplo 3:

(1) $\neg p \vee q \vee s$

(2) $\neg p \vee q$ $\qquad\qquad$ s

(3) $\neg p$ $\qquad$ q

Los tableaux tambien poseen algunas reglas, las cuales en el caso de que la formula a analizar contenga implicaciones, tenga negaciones como conectivos principales, etc., para evitarnos el trabajo de transformar la formula.

- α-reglas:
  1. De $A \wedge B$ se deduce A y B.
  2. De $\neg(A \vee B)$ se deduce $\neg A$ y $\neg B$.
  3. De $\neg(A \to B)$ se deduce A y $\neg B$.
- β-reglas:
  1. De $A \vee B$ se deduce A y, en una rama separada, B.
  2. De $\neg(A \wedge B)$ se deduce $\neg A$ y, en una rama separada, $\neg B$.

3. De $A \to B$ se deduce $\neg A$, y en una rama separada, B.

- σ-reglas:

1. De $\neg\neg A$ se deduce A.
2. De $\neg false$ se deduce $true$.
3. De $\neg true$ se deduce $false$.

$Nota$: No se dan reglas para la equivalencia ya que
$A \leftrightarrow B \equiv (A \to B) \wedge (B \to A) \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$

Ejemplos:

$$
\begin{array}{lcc}
(1) & \multicolumn{2}{c}{(p \to q) \to p} \\
(2) & \multicolumn{2}{c}{\neg(p \to q) \vee p} \\
(3) & \neg(p \to q) & p \\
(4) & p \wedge \neg q & \otimes \\
(5) & q & \\
 & \otimes &
\end{array}
$$

$$
\begin{array}{lcccccc}
(1) & & & & & \multicolumn{2}{c}{(p \to q) \wedge (r \to s) \wedge (\neg q \vee \neg s) \to \neg p \vee \neg r} \\
(2) & & & & \multicolumn{2}{c}{(p \to q) \wedge (r \to s) \wedge (\neg q \vee \neg s) \to \neg p} & \neg r \\
(3) & & & \multicolumn{2}{c}{\neg((p \to q) \wedge (r \to s) \wedge (\neg q \vee \neg s))} & \neg p & \\
(4) & \multicolumn{2}{c}{\neg(p \to q) \vee \neg(r \to s)} & \neg(\neg q \vee \neg s) & & & \\
(5) & \neg(p \to q) & \neg(r \to s) & & & & \\
(6) & p & r \to \neg s & & & & \\
(7) & \neg q & & & & & \\
(8) & & & q & & & \\
(9) & & & s & & &
\end{array}
$$

Ahora para determinar si si la formula es satisfacible o no, primero nos fijamos en que ramas se cierran, esto ocurre cuando sobre el mismo camino, encontramos una literal y su complementaria. Tenemos los siguientes casos:

- Si cierran todas las ramas entonces la formula es una negación y es insatisfacible.

  Ejemplo:

$$
\begin{array}{ll}
(1) & \neg p \wedge ((\neg r \wedge q) \wedge r) \\
(2) & p \\
(3) & r \wedge \neg q) \wedge r \\
(4) & r \neg q \\
(5) & \neg r \\
(6) & \neg r \\
 & \otimes
\end{array}
$$

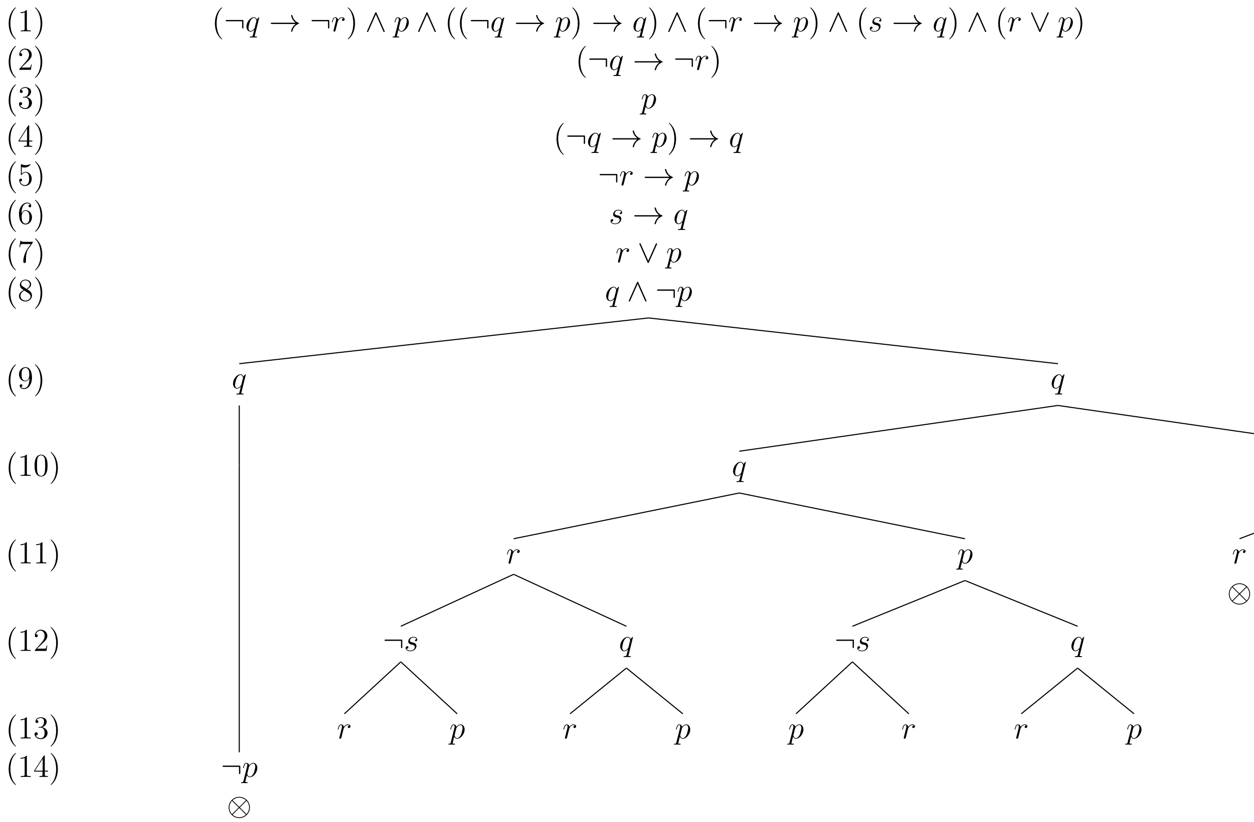
- Para saber si es una tautologia y/o si el argumento es correcoto negamos la formula y al aplicar el tableux se cerraran todas las ramas
  EJEMPLO: Tenemos la expresión $((p \rightarrow q)p) \rightarrow (q \rightarrow p)$ la cual es equivalente a $(\neg p \vee q) \wedge p) \vee (\neg q \vee p)$ y ya que para este caso queremos saber si es tautologia, negamos la expresion y obtenemos: $((p \wedge \neg q) \vee p) \wedge (q \wedge \neg p)$ y la analizamos con el tableux.

$$
\begin{array}{lccc}
(1) & & ((p \wedge \neg q) \vee p) \wedge (q \wedge \neg p) & \\
(2) & & q \wedge \neg p & \\
(3) & & (p \wedge \neg q) \vee p & \\
(4) & & q & \\
(5) & & \neg p & \\
(6) & p \wedge q & & p \\
 & & & \otimes \\
(7) & p \quad\quad \neg q & & \\
 & \otimes \quad\quad \otimes & &
\end{array}
$$

Como notamos se cierran todas las ramas, por lo tanto es una tautologia.

- Si quedan una o más ramas abiertas cada una es una posible interpretación donde las literales sin negación se toman como 1 y las negadas como 0.
  Ejemplo:
  Si tenemos el siguiente conjunto de formulas

$\Gamma = \{\neg q \to \neg r, p, (\neg q \to p) \to q, \neg r \to p, s \to q, r \vee p\}$
y queremos determinar si son satisfacibles, las unimos todas con ANDs
y las ecaluamos con el tableux.



Como se puede ver quedan varias ramas abiertas escogemos la que queramos, en este caso sera la primera de izquierda a derecha que queda abierta la cual nos da la siguiente interpretación tesultante: $\mathcal{I}(q) = 1$, $\mathcal{I}(r) = 1$, $\mathcal{I}(q) = 1$, $\mathcal{I}(q) = 0$,

Bibliografia

- Miranda, F., Viso, E.. (2016). Matemáticas Discretas 2a. ed.. México, UNAM.: Las prensas de Ciencias.

- Notas de clase.