Lógica Computacional
Nota 05. Resolución binaria*

Noé Salomón Hernández S.

1. Introducción

El método de resolución encuentra de manera eficiente una demostración de una fórmula a partir de las premisas mediante la regla de resolución, con la cual es posible construir un demostrador de teoremas que sea correcto y completo para la lógica proposicional. Las ventajas de este método serán aparentes hasta que se extienda a la lógica de primer orden.

La regla de resolución sólo se puede aplicar en expresiones en forma clausular. Antes de la aplicación, las premisas y la conclusión deben convertirse a esta forma. Afortunadamente, hay un proceso sencillo para realizar esta transformación.

Definición 1.1

- *Una* literal *es una proposición atómica o la negación de una proposición atómica.*
- *Sea ℓ una literal, ℓ y ℓ^c son literales complementarias, donde*

$$\ell^c = \begin{cases} p & si\ \ell = \neg p \\ \neg p & si\ \ell = p \end{cases}$$

- *Una* oración clausular *es una literal o una disyunción de literales.*
- *Una* cláusula *es el conjunto de literales en una oración clausular. Observe que el conjunto vacío de literales es una cláusula y equivale a una disyunción vacía; por lo tanto, es insatisfacible, se denota como $\Box$, y es un caso especial particularmente importante.*
- *La* cláusula unitaria *es una cláusula que consiste en una sola literal.*
- *Una fórmula en* forma clausular *es un conjunto de cláusulas. Implícitamente, esto indica que la forma clausular es una conjunción de cláusulas.*

Teorema 1.2 *Toda fórmula de la lógica proposicional puede ser transformada en una fórmula equivalente en forma clausular.*

Demostración. Para convertir una fórmula a su forma clausular se llevan a cabo los siguientes pasos, cada uno de los cuales preserva la equivalencia lógica.

*El material aquí presentado se basa el libro de Ben-Ari M., *Mathematical Logic for Computer Science.*

1. Eliminar implicaciones y bicondicionales al sustituirlas por la fórmulas equivalentes.

$$\begin{array}{rcl} \varphi \to \psi & \equiv & \neg\varphi \lor \psi \\ \varphi \leftrightarrow \psi & \equiv & (\neg\varphi \lor \psi) \land (\varphi \lor \neg\psi) \end{array}$$

2. Trabajar con la negación de modo que se aplique únicamente sobre proposiciones atómicas atómicas.

$$\begin{array}{lcl} \neg\neg\varphi & \equiv & \varphi \\ \neg(\varphi \land \psi) & \equiv & \neg\varphi \lor \neg\psi \\ \neg(\varphi \lor \psi) & \equiv & \neg\varphi \land \neg\psi \end{array}$$

3. Aplicar las leyes de la distributividad para eliminar conjunciones dentro de disyunciones.

$$\begin{array}{rcl} \varphi \lor (\psi \land \chi) & \equiv & (\varphi \lor \psi) \land (\varphi \lor \chi) \\ (\varphi \land \psi) \lor \chi & \equiv & (\varphi \lor \chi) \land (\psi \lor \chi) \end{array}$$

4. Ocupar la notación para la forma clausular.

$$\begin{array}{rcl} \varphi_1 \lor \ldots \lor \varphi_n & \rightsquigarrow & \{\varphi_1, \ldots, \varphi_n\} \\ \psi_1 \land \ldots \land \psi_m & \rightsquigarrow & \{\psi_1, \ldots, \psi_m\} \end{array}$$

El último paso de la demostración genera conjuntos, lo que causa que múltiples presencias de una cláusula se colapsen en una sola. La equivalencia lógica se preserva por idempotencia: $\varphi \land \varphi \equiv \varphi$ y $\varphi \lor \varphi \equiv \varphi$. $\dashv$

1.1. Cláusula trivial

Definición 1.3 *Una cláusula es* trivial *si contiene un par de literales complementarias. Se llama así porque trivialmente se satisface.*

Lema 1.4 *Sea S un conjunto de cláusulas y sea $C \in S$ una cláusula trivial. Entonces $S - \{C\}$ es lógicamente equivalente a S.*

Por lo tanto, asumiremos que todas las cláusulas triviales han sido eliminadas de las fórmulas en forma clausular.

Ejercicios 1.5

- Encuentra la Forma Normal Clausular para las siguientes fórmulas:
  - $(\neg p \to \neg q) \to (p \to q)$
  - $\neg(\neg q \leftrightarrow r)$
  - $p \land q \to r \lor s$
  - $p \land q \leftrightarrow r$
- Decidir si la fórmula $(p \lor q) \leftrightarrow (\neg p \to q)$ es válida utilizando la transformación a la Forma Normal Clausular.

2. Resolución

La resolución binaria proporciona un método de decisión para la lógica que utiliza el principio de refutación. El procedimiento de resolución consiste en una secuencia de aplicaciones de la regla de resolución sobre un conjunto de cláusulas. La regla preserva satisfacibilidad: si un conjunto de cláusulas es satisfacible, lo es también el conjunto de cláusulas que se genera por la aplicación de la regla. Por lo tanto, si la cláusula vacía (insatisfacible) se llega a obtener, el conjunto original de cláusulas debe ser insatisfacible.

Definición 2.1 (Regla de resolución) *Sean C_1 y C_2 dos cláusulas con $\ell \in C_1$ y $\ell^c \in C_2$. Decimos que C_1 y C_2 son cláusulas de colisión y que colisionan en el par de literales complementarias ℓ y ℓ^c. Obtenemos la cláusula C, el resolvente de C_1 y C_2, como sigue:*

$$C = \mathsf{Res}(C_1, C_2) = (C_1 - \{\ell\}) \cup (C_2 - \{\ell^c\}).$$

Podemos expresar esta relación al estilo de una regla de inferencia del siguiente modo,

$$\frac{C_1 \qquad C_2}{(C_1 - \{\ell\}) \cup (C_2 - \{\ell^c\})}\ (\mathsf{Res})$$

C_1 y C_2 son las cláusulas padre de C.

Es importante mencionar que las cláusula vacía o cláusula insatisfacible, $\square$, se obtiene como resultado de $\mathsf{Res}(\{\ell\}, \{\ell^c\})$.

La regla opera sobre un par de literales complementaras **a la vez** y **nunca con más**.

Lema 2.2 *Una disyunción de literales $\ell_1 \vee \ldots \vee \ell_m$ es válida syss existen $1 \leq i, j \leq m$ tal que ℓ_i es $\neg \ell_j$.*

Lema 2.3 *Si dos cláusulas colisionan en más de una literal, su resolvente es una cláusula trivial.*

Teorema 2.4 *El resolvente C es satisfacible syss las cláusulas padre C_1 y C_2 son ambas satisfacibles.*

Demostración. $\boxed{\rightarrow}$ Sea I una interpretación que satisface a C, entonces $I(\ell^\diamond) = 1$ para al menos alguna literal en $\ell^\diamond \in C$. Por la regla de resolución, $\ell^\diamond \in C_1$ o $\ell^\diamond \in C_2$, o en ambos. Si $\ell^\diamond \in C_1$, $I(C_1) = 1$. Ahora, $\ell \notin C$ ni $\ell^c \notin C$, por lo que I no está definida en ℓ o ℓ^c, así que extendemos I como I' definiendo $I'(\ell^c) = 1$. Como $\ell^c \in C_2$, tenemos $I'(C_2) = 1$; además $I'(C_1) = I(C_1) = 1$. Por lo tanto, I' es un modelo para C_1 y C_2. Análogamente cuando $\ell^\diamond \in C_2$.

$\boxed{\leftarrow}$ Sean C_1 y C_2 satisfacibles por la interpretación I. Como ℓ y ℓ^c son complementarias, se tiene que $I(\ell) = 1$, o bien, $I(\ell^c) = 1$. Supongamos que $I(\ell) = 1$, entonces $I(\ell^c) = 0$ y C_2, la cláusula que contiene a ℓ^2, se satisface porque hay otra literal $\ell' \in C$ ($\ell' \neq \ell^c$) tal que $I(\ell') = 1$. Por construcción de la regla de resolución, $\ell' \in C$, de modo que I también es un modelo de C. Análogamente cuando $I(\ell^c) = 1$.

Lema 2.5 *La cláusula vacía, $\square$, es insatisfacible. El conjunto vacío de cláusulas, $\varnothing$, es válido.*

Demostración. Una cláusula es satisfacible syss hay *alguna* interpretación bajo la cual *al menos una literal* en la cláusula es verdadera. Sea I una interpretación arbitraria. Como no hay literales en $\square$, no hay literales cuyo valor sea verdadero bajo I. Pero I es una interpretación arbitraria; por consiguiente, $\square$ es insatisfacible.

También se puede ver a $\square$ como el resolvente de las cláusulas padre $\{\ell\}$ y $\{\ell^c\}$. Como el teorema anterior establece, $\square$ es satisfacible syss $\{\ell\}$ y $\{\ell^c\}$ son satisfacibles, pero no se puede satisfacer a un par de literales complementarias. Por lo tanto, $\square$ es insatisfacible.

Un conjunto de cláusulas es satisfacible syss *toda* cláusula en él es verdadera para cualquier interpretación. Como no hay cláusulas en $\varnothing$ que requieran ser verdaderas, entonces $\varnothing$ es válido. $\dashv$

2.1. Algoritmo de resolución binaria

Algoritmo (Resolución binaria)
Input: Un conjunto de cláusulas S.
Output: S es satisfacible o insatisfacible.

Sea S un conjunto de cláusulas. Definimos $S_0 = S$.

Repetir los siguientes pasos para obtener S_{i+1} a partir de S_i:

- Escoja un par de cláusulas de colisión $\{C_1, C_2\} \subseteq S_i$ que no se hayan escogido anteriormente.
- Encuentre el resolvente $C = \mathsf{Res}(C_1, C_2)$ de acuerdo a la regla de resolución.
- Si C no es una cláusula trivial, definimos $S_{i+1} = S_i \cup \{C\}$; de otro modo, $S_{i+1} = S_i$.

Hasta que $C = \square$, en cuyo caso S es insatisfacible; o hasta que la regla de resolución se haya aplicado a todos los pares de cláusulas de colisión, lo que indica que S es satisfacible.

Ejercicios 2.6

- Decida si los siguientes argumentos lógicos son correctos por resolución binaria.
  - $\{\ell \vee m \to h, \neg(\neg h \wedge \neg m)\} \models h$
  - $\{p \vee (q \wedge r),\ p \to q,\ q \leftrightarrow s\} \models q \wedge s$
  - $\{p \to q,\ r \vee s \to t,\ t \to p\} \models p \to (\neg r \wedge s)$
  - Una condición necesaria para que llueva es que sea primavera o invierno. Si hay ríos, entonces llueve. Si es invierno, la gente utiliza gorros azules. La gente no utiliza gorros azules y hay ríos. Por lo tanto, es primavera. Utilice las claves:

    ℓ : llueve
    p : es primavera
    i : es invierno
    r : hay ríos
    a : la gente utiliza gorros azules

- Determine si los siguientes conjuntos son satisfacibles o no.
  - $\{p \wedge \neg(q \to s),\ p \vee q,\ s \leftrightarrow \neg p\}$
  - $\{p,\ p \to \big((q \vee r) \wedge \neg(q \wedge r)\big),\ p \to \big((s \vee t) \wedge \neg(s \wedge t)\big),\ s \to q,\ \neg r \to t,\ t \to s\}$

2.2. Correctud y completud

Teorema 2.7 (Correctud) *Sea S un conjunto de cláusulas. Si existe una refutación por resolución para S, entonces S es insatisfacible.*

Teorema 2.8 (Completud) *Si un conjunto de cláusulas es insatisfacible, entonces la cláusula vacía $\square$ será derivada por el procedimiento de resolución.*

3. Ejemplo

Decida si $\{p \to q, r \to s, p \vee r, \neg(q \wedge s)\} \models (q \to p) \wedge (s \to r)$ es correcto por resolución binaria. Por el principio de refutación, vamos a trabajar con

$$\{p \to q, r \to s, p \vee r, \neg(q \wedge s), \neg((q \to p) \wedge (s \to r))\}$$

Primero pasamos las fórmulas anteriores a forma clausular.

$$\begin{aligned}
p \to q &\equiv \{\neg p, q\} \\
r \to s &\equiv \{\neg r, s\} \\
p \vee r &\equiv \{p, r\} \\
\neg(q \wedge s) &\equiv \{\neg q, \neg s\} \\
\neg\big((q \to p) \wedge (s \to r)\big) &\equiv \neg(\neg q \vee p) \vee \neg(\neg s \vee r) \\
&\equiv (q \wedge \neg p) \vee (s \wedge \neg r) \\
&\equiv \big((q \wedge \neg p) \vee s\big) \wedge \big((q \wedge \neg p) \vee \neg r\big) \\
&\equiv (q \vee s) \wedge (\neg p \vee s) \wedge (q \vee \neg r) \wedge (\neg p \vee \neg r) \\
&\equiv \{q, s\}, \{\neg p, s\}, \{q, \neg r\}, \{\neg p, \neg r\}
\end{aligned}$$

Aplicando resolución tenemos lo siguiente: En azul están todos los resolventes que obtenemos de aplicar resolución a las premisas. En rojo aquellos que ocupan de las premisas y de los resolventes en azul. Esta forma esquemática permite saber si ya se ha aplicado resolución a todos los pares de cláusulas de colisión. Ahora se encuentran resolventes entre las premisas y las cláusulas en rojo, lo cual produce la cláusula 29, y finalmente llegamos a la cláusula vacía en la 30.

1.	$\neg p, q$	Prem
2.	$\neg r, s$	Prem
3.	p, r	Prem
4.	$\neg q, \neg s$	Prem
5.	q, s	Prem
6.	$\neg p, s$	Prem
7.	$q, \neg r$	Prem
8.	$\neg p, \neg r$	Prem
9.	q, r	$\text{Res}(1, 3)$
10.	$\neg p, \neg s$	$\text{Res}(1, 4)$
11.	s, p	$\text{Res}(2, 3)$
12.	$\neg r, \neg q$	$\text{Res}(2, 4)$
13.	r, s	$\text{Res}(3, 6)$
14.	p, q	$\text{Res}(3, 7)$
15.	$\neg q, \neg p$	$\text{Res}(4, 6)$
16.	$\neg s, \neg r$	$\text{Res}(4, 7)$
17.	q	$\text{Res}(1, 14)$
18.	$\neg p$	$\text{Res}(1, 15)$
19.	s	$\text{Res}(2, 13)$
20.	$\neg r$	$\text{Res}(2, 16)$
21.	$r, \neg s$	$\text{Res}(3, 10)$
22.	$p, \neg q$	$\text{Res}(3, 12)$
23.	$r, \neg q$	$\text{Res}(3, 15)$
24.	$p, \neg s$	$\text{Res}(3, 16)$
25.	$\neg p, r$	$\text{Res}(9, 15)$
26.	$q, \neg s$	$\text{Res}(9, 16)$
27.	$\neg q, s$	$\text{Res}(11, 15)$
28.	$p, \neg r$	$\text{Res}(11, 16)$
29.	p	$\text{Res}(3, 20)$
30.	$\square$	$\text{Res}(18, 29)$

Como se obtuvo la cláusula vacía, el conjunto de fórmulas es insatisfacible. Por consiguiente, el argumento es correcto.