Secuentes y el sistema de Gentzen G'
Lógica Matemática

José de Jesús Lavalle Martínez

14 de julio de 2011

Resumen

Este documento se basa en las ideas presentes en la sección *3.4 Proof Theory of Propositional Logic: The Gentzen System G'* del libro *Logic for Computer Science: Foundations of Automatic Theorem Proving* de Jean Gallier[1].

Primero definiremos formalmente el lenguaje de la lógica proposicional [1][2]. Para ello debemos definir su alfabeto y dar sus reglas sintácticas de formación.

Definición 0.1. El alfabeto del lenguaje de la lógica proposicional está conformado por:

1. Paréntesis: (,);

2. Conectivos lógicos: $\neg$, $\wedge$, $\vee$, $\rightarrow$;

3. Letras proposicionales: Cualquier letra con o sin subíndices.

Definición 0.2. Los elementos del lenguaje de la lógica proposicional, que llamaremos proposiciones, se definen mediante:

1. Toda letra proposicional es una proposición,

2. Si $\mathcal{A}$ y $\mathcal{B}$ son proposiciones entonces $(\neg\mathcal{A})$, $(\mathcal{A}\wedge\mathcal{B})$, $(\mathcal{A}\vee\mathcal{B})$ y $(\mathcal{A}\rightarrow\mathcal{B})$ también son proposiciones.

El sistema de Gentzen [1][2] nos permite saber si una proposición es una tautología, en caso de que no sea una tautología también nos permite saber bajo que asignaciones, de valores de verdad a las letras proposicionales, la proposición es falsificable, se basa en el concepto de secuente. Un secuente es una pareja (Γ, Δ) de secuencias (o conjuntos, posiblemente vacíos) de proposiciones: $\Gamma =< A_1, ..., A_n >, \Delta =< B_1, ..., B_m >$ y decimos que Γ y Δ son el antecedente y el consecuente, respectivamente. Por brevedad escribiremos los secuentes como $\Gamma \rightsquigarrow \Delta$ y en las secuencias no usaremos paréntesis angulares.

Las reglas de inferencia están formadas como un conjunto de secuentes premisas sobre una raya horizontal y un único secuente conclusión bajo la raya.

$$\frac{premisa_1, premisa_2, \ldots, premisa_n}{conclusion}\ nombreDeLaRegla$$

Las premisas resultan de la aplicación de la definición de la regla a la conclusión. Las reglas de inferencia se clasifican en dos categorías: las que operan sobre el antecedente $(* \rightsquigarrow)$ y las que operan sobre el consecuente $(\rightsquigarrow *)$: cada regla descompone a la proposición principal en subproposiciones que son colocadas ya sea en el antecedente o consecuente de las premisas, pudiendo incluso dividir un secuente (conclusión) en dos (premisas). Las premisas obtenidas después de esta operación pueden contener o no operadores lógicos, en caso de contenerlos las reglas de inferencia vuelven a ser aplicadas. Ello naturalmente permite representar a las pruebas como árboles.

Definición 0.3. El sistema de Gentzen se define a continuación:

1. Γ, Δ representan secuencias arbitrarias de fórmulas (conjuntos),

2. A, B representan proposiciones arbitrarias,

3. El único axioma es:
$$\frac{}{\Gamma, A \rightsquigarrow A, \Delta}\ Ax$$

4. Las reglas de inferencia de Gentzen son:

$$\frac{\Gamma \rightsquigarrow A, \Delta}{\Gamma, (\neg A) \rightsquigarrow \Delta}\ \neg \rightsquigarrow \qquad \frac{\Gamma, A \rightsquigarrow \Delta}{\Gamma \rightsquigarrow (\neg A), \Delta}\ \rightsquigarrow \neg$$

$$\frac{\Gamma, A, B \rightsquigarrow \Delta}{\Gamma, (A \wedge B) \rightsquigarrow \Delta}\ \wedge \rightsquigarrow \qquad \frac{\Gamma \rightsquigarrow A, \Delta \quad \Gamma \rightsquigarrow B, \Delta}{\Gamma \rightsquigarrow (A \wedge B), \Delta}\ \rightsquigarrow \wedge$$

$$\frac{\Gamma, A \rightsquigarrow \Delta \quad \Gamma, B \rightsquigarrow \Delta}{\Gamma, (A \vee B) \rightsquigarrow \Delta}\ \vee \rightsquigarrow \qquad \frac{\Gamma \rightsquigarrow A, B, \Delta}{\Gamma \rightsquigarrow (A \vee B), \Delta}\ \rightsquigarrow \vee$$

$$\frac{\Gamma \rightsquigarrow A, \Delta \quad \Gamma, B \rightsquigarrow \Delta}{\Gamma, (A \to B) \rightsquigarrow \Delta}\ \to\rightsquigarrow \qquad \frac{\Gamma, A \rightsquigarrow B, \Delta}{\Gamma \rightsquigarrow (A \to B), \Delta}\ \rightsquigarrow\to$$

Note que en el sistema de Gentzen tenemos un axioma (Ax) y dos formas de reglas de inferencia, aquellas que dado un secuente conclusión obtenemos sólo un secuente premisa llamémosles coloquialmente las que no bifurcan y aquellas que dado un secuente conclusión obtenemos dos secuentes premisas llamémosles coloquialmente las que bifurcan.

Como se puede observar una instancia de nuestro único axioma es cualquier secuente $\Gamma \rightsquigarrow \Delta$ tal que $\Gamma \cap \Delta \neq \varnothing$, esto es, contienen alguna proposición en común.

Ejemplo 0.4. La proposición $((p \to q) \to ((\neg q) \to (\neg p)))$ es una tautología y su árbol de deducción es el siguiente:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\overline{p, q \rightsquigarrow q}\ Ax}{q \rightsquigarrow q, (\neg p)}\ \rightsquigarrow\neg \qquad \cfrac{\overline{p \rightsquigarrow p, q}\ Ax}{\rightsquigarrow p, q, (\neg p)}\ \rightsquigarrow\neg}{(p \to q) \rightsquigarrow q, (\neg p)}\ \to\rightsquigarrow}{(\neg q), (p \to q) \rightsquigarrow (\neg p)}\ \neg\rightsquigarrow}{(p \to q) \rightsquigarrow ((\neg q) \to (\neg p))}\ \rightsquigarrow\to}{\rightsquigarrow ((p \to q) \to ((\neg q) \to (\neg p)))}\ \rightsquigarrow\to$$

Ejemplo 0.5. De la misma forma si queremos saber si la proposición $((\neg p) \wedge p)$ es siempre verdadera, podemos construir un árbol de deducción para el secuente $\rightsquigarrow ((\neg p) \wedge p)$. De tal manera encontramos que la proposición $((\neg p) \wedge p)$ no es una tautología y se puede falsificar cuando p toma el valor veritativo verdadero (aunque también cuando p es falso, ya que en realidad es una contradicción). Su árbol de deducción es el siguiente:

$$\dfrac{\dfrac{p \rightsquigarrow}{\rightsquigarrow (\neg p)}\ \rightsquigarrow \neg \qquad \rightsquigarrow p}{\rightsquigarrow ((\neg p) \wedge p)}\ \rightsquigarrow \wedge$$

Referencias

[1] Jean Gallier; Logic for Computer Science: Foundations of Automatic Theorem Proving; 2003, `http://www.cis.upenn.edu/~jean/gbooks/logic.html`, last visited July 2011.

[2] Steve Reeves and Mike Clarke; Logic for Computer Science, Addison-Wesley Publishers Ltd. 1990, `http://www.cs.waikato.ac.nz/~stever/LCS.pdf`, last visited July 2011.