SISTEMA EXPERTO EN DEDUCCIÓN NATURAL

Gabriel Garduño Soto, [1]
David René Thierry García,[2]
Rafael Vidal Uribe,[1]
Hugo Padilla Chacón[3]

1. Panorama actual del problema de la deducción natural

El problema de la deducción natural ha sido abordado desde distintas perspectivas. En la mayor parte de las investigaciones realizadas hasta la fecha se han intentado soluciones que, con base en un enfoque sintáctico, tratan de probar la validez de las conclusiones propuestas a los diversos grupos de premisas objeto de investigación. En dichos enfoques sintácticos, se utilizan técnicas de recorrido lineal, o arborescente, tal como el *backtracking*, o bien, búsquedas heurísticas cuya mayor dificultad es el enorme gasto en tiempo y memoria de máquina empleado. Todo usuario del lenguaje *Prolog* conoce el constante riesgo de saturación de la memoria RAM, a menos de que se controlen estrictamente la emisión de reglas de producción (o transformaciones sintácticas) y el recorrido global sobre cada una de las premisas declaradas, mediante los comandos de corte incorporados en *Prolog*.

De este modo, se han publicado trabajos con resultados alentadores, pero incompletos, para la investigación del problema de la deducción natural; sirvan a manera de ejemplo los enfoques siguientes:

- Métodos gráficos de distribución hipergeométrica (Urquhart 1987).
- Métodos algorítmicos de recorrido lineal sobre todas las cláusulas de Horn que se pueden producir a partir del conjunto de premisas propuesto (Arvind 1987) y métodos de chequeo clausular con fuerza bruta, a través de enfoques en paralelo (Chen 1987).
- Métodos de búsqueda arborescente (Amir 1987).
- Métodos de investigación matemática sobre anillos booleanos (Nakao 1986).
- Métodos de investigación sobre el tiempo de máquina requerido para dilucidar la validez de la conclusión propuesta (Walton 1987).

En el trabajo aquí presentado, se ha implementado una aritmetización completa de la lógica bivalente que bien podría denominarse lógica aritmetizada sin rastreo (*backtracking*), pues los algoritmos propuestos permiten un acercamiento directo y prácticamente sin gasto de tiempo de máquina en la búsqueda heurística, ya que en su indagación acerca de la validez de una conclusión, el sistema se dirige infaliblemente al lugar matemático en donde se produce dicha conclusión, en caso de ser válida; en el caso contrario, el sistema informa acerca de su no-validez.

En la mayor parte de los intentos que se han realizado para resolver el problema de la deducción natural se ha formalizado a la lógica desde distintas perspectivas, pero formalizar no es lo mismo que aritmetizar. Toda aritmetización de la lógica es, *eo ipso*, una formalización; la inversa no siempre es válida. La relación no es necesariamente simétrica.

2. Breves consideraciones sobre tres aritmetizaciones de la lógica

Es bien conocido el recurso fundamental de que se sirve Gödel para desarrollar su teorema de incompletud (1931): aritmetiza el nivel de la lógica. A cada signo elemental, a cada fórmula y a cada secuencia de fórmulas de la lógica les asigna un número único (número de Gödel) de manera que la lógica queda "mapeada" en la aritmética. Este mapeo en Gödel no es un fin en sí mismo: es sólo un recurso, entre otros, de que se vale para alcanzar el propósito de probar el teorema. Quizá por ello no fue óbice que el mapeo no resultara biunívoco, es decir, que mientras a cada expresión o secuencia de expresiones lógicas les corresponde un único número (de Gödel), no todo número es un número de Gödel y, por lo tanto, no todo número representa una expresión o secuencia de expresiones lógicas. Por esto la aritmetización de Gödel no permite ir más allá, desde un punto de vista operativo.

1 Facultad de Filosofía y Letras, UNAM. Colegio de Filosofía, División SUAFyL.

2 Facultad de Economía, UNAM. División SUAE.

3 Facultad de Filosofía y Letras, UNAM. División de Estudios de Posgrado.

En diversos trabajos, Łukasiewicz empleó recursos de aritmetización para tratar problemas de lógica. Sin embargo, la aritmetización misma tampoco la tuvo como fin en sus investigaciones. A diferencia de Gödel que estableció un mecanismo estricto para garantizar la univocidad (aunque no la biunivocidad, como se dijo antes), Łukasiewicz siempre estableció convenciones laxas y coyunturales, de manera que los números empleados, aunque reflejan ciertas estructuras profundas de la lógica, tampoco permiten una operatividad sistemática.

En el método de Quine-Mcklusky para minimizar funciones booleanas canónicas, también puede encontrarse una incipiente aritmetización. Tampoco es un objetivo en sí misma y sólo auxilia, dentro del propósito del método, al fin de agrupar los sumandos lógicos potencialmente minimizables.

No hemos encontrado una aritmetización igual a la que ahora se expone en sus rasgos generales.

3. Aritmetización del cálculo proposicional

3.1 La aritmetización que hemos desarrollado está constituida por un grupo de algoritmos que resuelven como problemas estrictamente aritméticos los problemas del cálculo proposicional.

3.2 Las reglas de correspondencia permiten garantizar una biunivocidad, en el siguiente sentido: a toda y a cada fórmula del cálculo proposicional le corresponde uno y sólo un número; a todo y a cada número le corresponde una y sólo una fórmula del cálculo proposicional.

3.3 Los números constituyen los valores que pueden ser sustituidos por las variables, parámetros y constantes en los algoritmos. Los números que resultan de realizar las operaciones aritméticas señaladas en los algoritmos representan la solución parcial o total de un problema lógico parcial o total.

3.4 Una vez resuelto un problema lógico por medio de los algoritmos, en virtud de 3.2 (biunivocidad) la solución aritmética se reinterpreta en notación lógica.

3.5 La validez de los algoritmos fue probada por medio de la inducción matemática completa.

Como consecuencia de la aritmetización lograda, puede establecerse una conexión con el conocido teorema de Shannon (1938). El teorema de Shannon establece una biunivocidad entre las fórmulas del álgebra booleana y los circuitos en serie-paralelo: cada fórmula del álgebra booleana es realizable como un circuito en serie-paralelo; a la inversa, cada circuito en serie-paralelo es representable como una fórmula del álgebra booleana. Ahora puede postularse una extensión de este teorema: a cada número le corresponde una fórmula booleana canónica y, puesto que (por el teorema de Shannon) a cada fórmula booleana le corresponde un circuito en serie-paralelo, entonces a cada número le corresponde un circuito en serie-paralelo. La inversa es válida también.

4. Sistema para resolver problemas de deducción

Los algoritmos mencionados conforman un sistema en el sentido en que grupos y subgrupos de los mismos están destinados a resolver problemas específicos, pero todos ellos son interrelacionables y, por esto, permiten resolver problemas con mayor grado de complejidad. De esta manera, el sistema, al que denominaremos, *SD*, puede ser definido como:

$$SD = < \{A_1, A_2, \ldots, A_n\}, \{0, 1, 2, \ldots, n\}, R >$$

en donde $\{A_1, A_2, \ldots, A_n\}$ es el conjunto de los algoritmos; $\{0, 1, 2, \ldots, n\}$ es el conjunto de los valores que pueden cobrar las variables y constantes de los algoritmos, y R es una relación que permite la interconexión entre los algoritmos.

El sistema que conforman los algoritmos es experto en el sentido en que puede "simular" la conducta de un lógico especializado en problemas de deducción dentro del cálculo proposicional, aunque también dentro del cálculo cuantificacional bajo ciertas restricciones. La silogística aristotélica, por ejemplo, aunque perteneciente por tradición a una lógica de "términos", es manejable dentro del sistema. El sistema es experto aún en el sentido psicológico en que el sistema puede dar "sorpresas", inclusive a sus propios creadores. A manera de ilustración: cuando estaba siendo sometido a ensayos y provisto de información concerniente a la silogística clásica, "reportaba" como inválidos varios silogismos incluidos como válidos en algunos listados estándar: *Bamalip*, *Darapti*, etc. Se pensó, en primera y más grave instancia, en un error en los algoritmos; luego, en un error en la interconexión; luego, en un error en la información, etc. No había tal: el sistema estaba bien. Los silogismos que rechazaba son dubitables en tanto que

silogismos genuinos, puesto que implican una presunción de existencia no contenida en las premisas universales. Para percatarse de esto, hubo necesidad de revisar las discusiones sobre la silogística clásica. Los creadores del sistema, supuestos expertos, habían olvidado estas cuestiones.

No es posible ofrecer, en la reducida extensión a que están constreñidos los trabajos que se presentan en una conferencia, los algoritmos que conforman el sistema. Su exposición y explicación consumirían un tiempo equivalente, por lo menos, al de un curso con duración de dos semestres académicos. No son muy numerosos, son aproximadamente cien. Tampoco son difíciles de comprender, sólo se requiere el manejo de dos teorías elementales, la de la lógica y la de los números. Pero algunos de ellos son extensos y aparentemente complejos. No obstante, se destacarán en seguida algunas de sus características fundamentales:

4.1. Permiten distinguir entre las fórmulas bien formadas (*wffs*) y las simples expresiones.

4.2. Permiten saber si en un conjunto de *wffs* consistentes, sintácticamente diferentes, una o más fórmulas son redundantes, esto es, si son semánticamente equivalentes; o bien, si la última de las *wffs* es redundante por deductibilidad respecto de las *wffs* anteriores.

4.3. Permiten transformar cualquier fórmula bien formada, con cualquier número y combinación de los operadores lógicos clásicos, en una fórmula canónica (normal disyuntiva).

4.4. Permiten decidir si una fórmula determinada cualquiera, con cualquier número y combinación de los operadores lógicos clásicos, es o no un teorema de un cálculo proposicional axiomatizado.

4.5. Permiten decidir, en deducción natural, si una fórmula determinada se deduce o no de un conjunto de premisas dado.

4.6. Permiten obtener, en deducción natural, **todas** las conclusiones semánticamente distintas que se deducen de un conjunto de premisas dado.

4.7. Resuelven un problema inusual dentro de la lógica. A saber: permiten obtener, en deducción natural, **todos** los conjuntos semánticamente distintos de premisas de los cuales se deduce –una vez determinado el número de variables en que se quiera encuadrar el problema– una fórmula cualquiera propuesta a manera de conclusión.

4.8. Los problemas se pueden plantear con cualquier número y combinación de los operadores lógicos tradicionales; las soluciones, en los casos 4.6 y 4.7, se obtienen en forma normal disyuntiva.

Con la adición de dos subsistemas más, cuyos algoritmos están en un avanzado proceso de elaboración, en un futuro cercano será posible: *a*) obtener las fórmulas canónicas en forma minimizada y *b*) mediante una programación de gráficas, obtener directamente el diseño del circuito en serie-paralelo minimizado, que corresponda a un determinado problema lógico planteado.

5. Sistema y Programación

Los algoritmos contenidos en el sistema permiten trabajar, en principio, con cualquier número finito de variables proposicionales, tan grande como se quiera. Sin embargo, las limitaciones propias de los equipos de cómputo personales (PC), restringen su efectividad, por ahora, a un máximo de cuatro variables. No obstante esto, se producen resultados interesantes: hay planteamientos, en el caso 4.6, en que se llegan a obtener más de 30,000 conclusiones semánticamente distintas, a partir de un conjunto de premisas con cuatro variables.

6. Acotación teórica

La filosofía subyacente en el desarrollo de los algoritmos es eminentemente constructivista. Por esto, la línea de investigación se encuentra cercana a los enfoques del intuicionismo: Krönecker, Poincaré, Borel, Brower, Weyl, Heyting, etc., aunque esto no significa que se asuman todas las tesis de esta corriente.

Mayo de 1990

Referencias:

- Amir A.: "Expressive Completeness Failure in Branching Time Structures". *Journal of Computer and System Sciences* 34, (1), 27-42, 1987.
- Arvind V., Biswas S.: "An O(n^2) Algorithm for the Satisfiability Problem of a Subset of Propositional Sentences in CNF that Includes all Horn Sentences". *Information Processing Letters* 24, (1), 67-69, 1987.
- Chen Wen-Tsuen, Liu Lung-Lung: "Parallel Approach for Theorem Proving in Propositional Logic". *Information Sciences* 41, (1), 61-76, Feb. 1987.
- Nakao Z.: "An Extension of Logic Functions on the Quaternary Boolean Ring". *The Transactions of the IECE of Japan, E* 69, (11),1169-1172, Nov. 1986.
- Urquhart A.: "Hard Examples for Resolution". *Journal of the Association for Computing Machinery*, 34, (1),209-219, 1987.
- Walton Purdom P.Jr., Brow A. C.: "Polynomial-Average-Time Satisfiability Problems". *Information Sciences* 41, (1), 23-42, 1987.

Nota editorial:

Este trabajo fue presentado en el *Ier . Coloquio en Ciencia Cognitiva*. Universidad de Guadalajara y Sociedad Filosófica Iberoamericana. Facultad de Filosofía y Letras de la Universidad de Guadalajara. Guadalajara. 19 y 20 de Julio de 1990.

Una versión preliminar de este documento fue publicada en: *Va. Confer encia Internacional: Las Computadoras en Instituciones de Educación y de Investigación*. Cómputo Académico UNAM, UNISYS, México. Noviembre 14 a Noviembre 16 de 1989.