Lógica Computacional, 2025-2
Unidad 1: Introducción

Sistemas Lógicos y sus Propiedades

Manuel Soto Romero

6 de febrero de 2025
Facultad de Ciencias UNAM

En esta nota daremos una definición detallada del concepto de sistema lógico, analizando sus componentes fundamentales: la sintaxis, que establece las reglas para formar fórmulas; la semántica, que asigna significado a dichas fórmulas; y las reglas de inferencia, que permiten derivar conclusiones a partir de premisas o axiomas. Asimismo, se revisan propiedades importantes, como la consistencia, completitud, corrección, decidibilidad, compactación, monotonicidad y clausura bajo operaciones lógicas, además de la relación entre decidibilidad y expresividad. Revisaremos también ejemplos y aplicaciones prácticas que ilustran la importancia de estos conceptos en áreas como la inteligencia artificial, la verificación formal y la optimización de algoritmos.

1. ¿Qué es un Sistema Lógico?

En el contexto de la lógica computacional y las ciencias de la computación, un sistema lógico es un marco formal que define reglas y una sintaxis específicas para representar y razonar sobre proposiciones y enunciados dentro de un dominio determinado. Estos sistemas establecen un conjunto de reglas formales y semánticas que permiten inferir conclusiones lógicas a partir de premisas o axiomas dados.

- **Sintaxis**
  Especifica el conjunto de símbolos y las reglas gramaticales que se utilizan para expresar proposiciones o fórmulas dentro del sistema. En esencia, define cómo combinar símbolos para construir fórmulas válidas.

- **Semántica**
  Proporciona el significado de las expresiones dentro del sistema lógico. Establece reglas que asignan interpretaciones a las fórmulas, permitiendo evaluar su veracidad y razonar sobre ellas.

- **Reglas de Inferencia**
  Son normas que permiten derivar nuevas fórmulas lógicas a partir de las existentes. Estas reglas determinan cómo obtener conclusiones correctas a partir de premisas o axiomas dados.

Los sistemas lógicos varían en su complejidad y capacidad de expresividad. Ejemplos comunes incluyen la lógica proposicional, la lógica de predicados (también conocida como lógica de primer orden) y la lógica de segundo orden, entre otros. Cada sistema se define por un conjunto específico de reglas sintácticas y semánticas, así como por reglas de inferencia que determinan cómo se llevan a cabo los razonamientos lógicos dentro de ese marco formal.

2. Propiedades de los Sistemas Lógicos

En el estudio de los sistemas lógicos, se analizan varias propiedades fundamentales que garantizan su utilidad, coherencia y capacidad para modelar razonamientos válidos. Estas propiedades permiten evaluar la corrección y aplicabilidad del sistema en distintos contextos, como la inteligencia artificial o la verificación de programas.

Consistencia

Un sistema lógico es consistente si nunca llegamos a conclusiones que se contradigan entre sí. En otras palabras, no es posible llegar a una situación donde afirmemos algo y también afirmemos lo opuesto al mismo tiempo.

Ejemplo 1

Si decimos *Hoy llueve* y también *Hoy no llueve*, estamos en una contradicción. Un sistema lógico consistente nunca permitirá que ambas cosas sean ciertas a la vez.

Esta propiedad es fundamental para garantizar que los sistemas computacionales basados en lógica no produzcan contradicciones. Si un sistema lógico fuera inconsistente, cualquier enunciado podría derivarse, lo que haría imposible distinguir razonamientos válidos de inválidos. Tiene aplicaciones en la verificación de programas, diseño de software confiable y razonamiento automatizado.

Completitud

Un sistema lógico es completo si todo lo que es verdadero según las reglas del *mundo lógico*[1] puede ser demostrado usando las reglas del sistema. En otras palabras, no existen verdades que el sistema no pueda justificar.

Ejemplo 2

El teorema de completitud de Gödel asegura que, en la lógica proposicional, cualquier afirmación que sea verdadera también puede ser demostrada con las reglas del sistema. Por ejemplo, si algo es siempre cierto como *si llueve, el suelo está mojado*, el sistema podrá demostrarlo.

Permite demostrar que todas las verdades dentro del sistema lógico pueden ser derivadas formalmente. Esto asegura que el sistema lógico es suficientemente potente para capturar todos los razonamientos válidos dentro de su dominio. Tiene aplicaciones en sistemas de prueba automática (e.g., PROLOG), garantizar que todas las conclusiones correctas puedan derivarse es esencial para resolver problemas complejos.

Corrección

Un sistema lógico es correcto si todo lo que podemos demostrar con sus reglas es realmente verdadero según las reglas del mundo lógico. En otras palabras, el sistema no se *equivoca* al permitirnos demostrar algo que no tiene sentido o no es cierto.

[1] En este contexto, el mundo lógico es el "escenario" donde comprobamos si las afirmaciones tienen sentido. Por ejemplo, si decimos "hoy llueve", el mundo lógico es ese entorno en el que, usando nuestras reglas, decidimos si esa afirmación se ajusta a la realidad o no.

Ejemplo 3

En lógica proposicional, si demostramos que *si hay lluvia y viento, entonces hay lluvia*, podemos estar seguros de que esto siempre será cierto en cualquier situación lógica válida.

- ⋆ **Vídeo Recomendado**: *Las Matemáticas son para Siempre - Eduardo Saenz de Cabezón*

Garantiza que todo lo que se derive (demuestre) en el sistema lógico sea verdadero en el modelo semántico. sto es importante para que los resultados obtenidos mediante razonamiento automático sean confiables. Tiene aplicaciones en sistemas como los solucionadores SAT o los solucionadores SMT que dependen de la corrección para ofrecer soluciones verificables a problemas complejos.

Decibidibilidad

Un sistema lógico es decidible si existe un método claro y automático para determinar, en un tiempo finito, si una afirmación es verdadera o falsa dentro del sistema. En otras palabras, siempre podemos encontrar una respuesta definitiva (sí o no) sobre si algo puede ser demostrado o no.

Ejemplo 4

En lógica proposicional, hay métodos (como tablas de verdad) que nos permiten comprobar, de forma segura y en un tiempo razonable, si una afirmación es válida o no. Por ejemplo, podemos verificar si *si llueve y hay viento, entonces hay viento* es verdadera o falsa usando estas herramientas.

La decidibilidad es importante en lógica computacional porque permite desarrollar algoritmos que determinen automáticamente si una fórmula es válida, satisfacible o insatisfacible en un tiempo finito. Tiene aplicaciones en la resolución de problemas de satisfacibilidad booleana (SAT), análisis de modelos, y herramientas de verificación de software.

Compactación

Un sistema lógico tiene la propiedad de compactación si, cuando un conjunto infinito de afirmaciones tiene una solución, también podemos encontrar esa solución considerando solo un subconjunto finito de esas afirmaciones. En otras palabras, para demostrar que algo es cierto, no necesitamos revisar infinitas afirmaciones; basta con analizar un número limitado de ellas.

Ejemplo 5

En lógica proposicional, si tenemos infinitas proposiciones que dicen *Hoy llueve o mañana llueve*, *Hoy llueve o pasado mañana llueve* y así sucesivamente, podemos demostrar que al menos una es verdadera revisando solo un subconjunto finito, como las primeras dos o tres afirmaciones.

Facilita el razonamiento lógico al reducir problemas infinitos a un subconjunto finito. Esto es esencial para que los sistemas computacionales puedan trabajar con recursos limitados. Tiene aplicación en optimización de algoritmos de resolución y razonamiento automatizado, como los utilizados en herramientas de verificación formal

Monotonicidad

Un sistema lógico tiene la propiedad de monotonicidad si agregar nuevas premisas nunca invalida las conclusiones que ya se habían demostrado. En otras palabras, si algo es cierto con las premisas que tenemos, seguirá siendo cierto aunque añadamos más información.

Ejemplo 6

En lógica proposicional, si demostramos que *si llueve, entonces el suelo está mojado* con la premisa *llueve*, esto seguirá siendo cierto incluso si añadimos la premisa *y además hace frío*. Las nuevas premisas no afectan lo que ya habíamos concluido.

En lógica clásica, la monotonicidad asegura que agregar más información a un conjunto de premisas no invalida conclusiones anteriores. Esto simplifica la acumulación de conocimiento en sistemas computacionales. Tiene aplicaciones en el razonamiento lógico dentro de bases de datos deductivas, planificación automática y sistemas de inferencia.

Clausura Bajo Operaciones Lógicas

Un sistema lógico tiene la propiedad de clausura bajo operaciones lógica si, al combinar afirmaciones con operaciones como y ($\wedge$), o ($\vee$) o no ($\neg$), las nuevas afirmaciones que se forman también pertenecen al sistema. En otras palabras, si empezamos con afirmaciones válidas dentro del sistema, las combinaciones que hagamos con las reglas lógicas seguirán siendo válidas.

Ejemplo 7

En lógica proposicional, si sabemos que *llueve* y que *hace viento* son afirmaciones válidad, también lo será la combinación *llueve y hace viento* ($p \wedge q$).

Decidibilidad vs. Expresividad

La relación entre decidibilidad y expresividad en un sistema lógico es un equilibrio: a medida que el sistema puede expresar ideas más complejas (expresividad), se vuelve más difícil garantizar que siempre podamos decidir si algo es verdadero o falso (decidibilidad). En otras palabras, los sistemas más expresivos pueden describir cosas más ricas y detalladas, pero es posible que no podamos resolver todo dentro de ellos de forma automática o segura.

Ejemplo 8

- ⋆ La lógica proposicional es decidible porque podemos usar métodos automáticos, como tablas de verdad, para comprobar cualquier afirmación. Sin embargo, su expresividad es limitada y no puede describir conceptos más complejos, como relaciones entre objetos.

- ⋆ La lógica de predicados es más expresiva, ya que puede hablar de relaciones como *Todos los gatos son mamíferos*, pero pierde decidibilidad en ciertos casos, ya que no siempre podemos garantizar una respuesta automática para todas las afirmaciones.

Existe un equilibrio entre la decidibilidad y la expresividad en lógica computacional. Los sistemas lógicos más expresivos (como la lógica de primer orden) tienden a ser indecidibles, mientras que sistemas más simples (como la lógica proposicional) son decidibles. Este equilibrio es crítico para elegir el sistema lógico adecuado para una aplicación específica. Tiene aplicaciones en la elección de la lógica adecuada en verificación de programas o diseño de sistemas inteligentes.

3. Conclusión

Los sistemas lógicos son herramientas esenciales para razonar de manera estructurada, definidos por su sintaxis, semántica y reglas de inferencia. Propiedades como consistencia, completitud, corrección y decidibilidad garantizan su confiabilidad, mientras que el equilibrio entre expresividad y decidibilidad asegura su aplicabilidad según el problema a resolver. Estas propiedades hacen de los sistemas lógicos una base relevante en áreas como la inteligencia artificial y la verificación de software.

Referencias

[1] Enderton, H. B. (2001). *A Mathematical Introduction to Logic* (2nd ed.). Elsevier.

[2] Gödel, K. (1931). *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*. Monatshefte für Mathematik und Physik, 38(1), 173–198.

[3] Mendelson, E. (2015). *Introduction to mathematical logic* (6th ed.). CRC Press.

[4] Shoenfield, J. R. (2001). *Mathematical logic*. A K Peters/CRC Press.

[5] Harrison, J. (2009). *Handbook of practical logic and automated reasoning*. Cambridge University Press.