Lógica Computacional, 2025-2
Unidad 3: Lógica de Predicados

Análisis de Argumentos Correctos

Manuel Soto Romero   Luis Fernando Loyola Cruz

16 de mayo de 2025
Facultad de Ciencias UNAM

En esta nota abordaremos una de las nociones más importantes de la lógica de predicados: la consecuencia lógica, entendida como el criterio formal que nos permite determinar cuándo un argumento es semánticamente correcto. Comenzaremos formalizando esta noción con base en interpretaciones y estados, y mostraremos cómo extender el análisis que ya conocemos de la lógica proposicional. A través de un ejemplo sencillo (pero representativo) veremos cómo verificar paso a paso si una conclusión se sigue necesariamente de un conjunto de fórmulas. Finalmente, exploraremos cómo automatizar este proceso utilizando funciones en Haskell, con el fin de construir herramientas que simulen el razonamiento lógico y faciliten el análisis de argumentos en entornos computacionales.

1. Análisis de Argumentos Correctos en Lógica de Predicados

La noción de argumento correcto se formaliza en la lógica de primer orden a través del concepto de consecuencia lógica, cuya definición es la misma que en la lógica proposicional.

Definicón 1. (Consecuencia Lógica)

*Sea Γ un conjunto de fórmulas y φ una fórmula individual. Diremos que φ es **consecuencia lóigca** de Γ, y lo escribimos como $\Gamma \vDash \varphi$, cuando se cumple que toda interpretación o modelo hace verdaderas a todas las fórmulas de Γ, también hace verdadera φ. En todas palabras, para todo modelo $\mathcal{M}$, si $\mathcal{M} \vDash \Gamma$, entonces también $\mathcal{M} \vDash \varphi$. En este caso, también se dice que Γ **implica lógicamente** a φ.*

Esta definición extiende a la lógica de predicados la noción de consecuencia lógica que ya conocemos de la lógica proposicional. En ese sistema más simple, decimos que una fórmula φ es consecuencia lógica de un conjunto de fórmulas Γ si, bajo todo estado que hace verdaderas a todas las fórmulas de Γ, la fórmula φ también resulta verdadera. Esta relación se analiza mediante interpretaciones o tablas de verdad.

En lógica de predicados, ya no basta con asignar valores de verdad a las fórmulas, porque el lenguaje es más expresivo: incluye constantes, funciones, predicados y cuantificadores. Por ello, el análisis semántico se realiza a través de interpretaciones, que especifican un universo no vacío y asignan significado a los símbolos del lenguaje. Decimos que una fórmula es verdadera bajo una interpretación si dicha interpretación satisface las condiciones semánticas de la fórmula.

Así, afirmamos que φ es consecuencia lógica de Γ (se escribe $\Gamma \vDash \varphi$) cuando toda interpretación que satisface todas las fórmulas de Γ, también satisface a φ. En otas palabras, la verdad de φ está garantizada por la verdad de las fórmulas en Γ en cualquier interpretación posible.

Esta definición mantiene la misma estructura que en la lógica proposicional, pero adaptada a un lenguaje más rico. Se trata, en el fondo, de la misma idea: la consecuencia lógica captura la corrección de un argumento desde el punto de vista semántico, es decir, de acuerdo al significado que le damos a las fórmulas a través de las interpretaciones.

Ejemplo 1

Vamos a demostrar que el siguiente argumento es **correcto** desde el punto de vista semántico:

Todos los gatos maúllan. Cleofas es un gato. Por lo tanto, Cleofas maúlla.

Formalización

La información del enunciado se representa en lógica de predicados de la siguiente manera:

- ⋆ Premisas:
$$\Gamma = \{\forall x(G(x) \to M(x)), G(c)\}$$
- ⋆ Conclusión:
$$M(c)$$

Queremos demostrar que la conclusión se sigue lógicamente de las premisas, es decir:

$$\Gamma \vDash M(c)$$

Supongamos que una interpretación $\mathcal{M}$ **satisface** las premisas de Γ, es decir:

$$\mathcal{M} \vDash \forall x(G(x) \to M(x)) \quad \text{y} \quad \mathcal{M} \vDash G(x)$$

Consideremos ahora un estado σ cualquiera.

De la satisfacción de la fórmula universal, se sigue que para todo objeto $m \in \mathcal{U}$, se tiene:

$$\mathcal{I}_{\sigma[x/m]}(G(x) \to M(x)) = 1$$

2. Automatización

Las nociones de la lógica de predicados, como la consecuencia lógica, pueden analizarse automáticamente mediante lenguajes funcionales como Haskell. En particular, es posible representar fórmulas, conjuntos de premisas y verificadores semánticos que comprueban si una conclusión se sigue de un conjunto de hipótesis bajo todas las interpretaciones posibles.

A continuación se muestran firmas de funciones en Haskell que permiten modelar este análisis:

```haskell
esValida :: Formula -> Interpretacion -> Estado -> Bool

satisface :: Interpretacion -> Estado -> ConjuntoFormula -> Bool

esConsecuenciaLogica :: ConjuntoFormula -> Formula -> Bool
```

Estas funciones permiten simular el comportamiento semántico de las interpretaciones y verificar si una fórmula es consecuencia lógica de un conjunto de fórmulas. Así el argumento formalizado como:

$$\Gamma = \{\forall x(G(x) \to M(x), G(c)\} \quad y \quad \varphi = M(c)$$

puede evaluarse automáticamente mediante:

```
esConsecuenciaLogica gamma phi
```

donde `gamma` representa las premisas y `phi` la conclusión. Este enfoque automatizado no sólo facilita la verificación de argumentos correctos, sino que también puede escalarse para asistir en tareas como generación de contraejemplos, diseño de sistemas de inferencia o soporte a la docencia en lógica computacional.

3. Conclusiones

En resumen, hemos aprendido a identificar cuándo una conclusión se sigue lógicamente de un conjunto de fórmulas en lógica de predicados, utilizando la noción formal de consecuencia lógica. Este concepto extiende de manera natural lo que ya conocíamos en lógica proposicional, pero requiere un análisis más profundo debido a la riqueza del lenguaje de primer orden. Además, exploramos cómo esta verificación puede automatizarse mediante programación funcional, lo cual no solo permite validar argumentos de manera eficiente, sino que abre la puerta al desarrollo de sistemas computacionales capaces de razonar formalmente. Esta conexión entre teoría y práctica es de gran importancia para comprender el papel de la lógica en las ciencias de la computación, particularmente dentro de la inteligencia artificial y la verificación formal.

Referencias

[1] Miranda Perea, F. E., Reyes, A. L., González, L. del C., & Linares, S. (2018). *Lógica Computacional: Notas de clase.* Facultad de Ciencias, Universidad Nacional Autónoma de México.

[2] Loyola Cruz, L. F. (2023). *Manual de Prácticas para la Asignatura de Lógica Computacional* (Proyecto de Apoyo a la Docencia). Universidad Nacional Autónoma de México.

[3] Reyes, A. L., & Reyes, R. (2021). *Matemáticas Discretas: Notas de clase.* Colegio de Ciencia y Tecnología, UACM San Lorenzo.

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

[5] Huth, M., & Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning about Systems* (2nd ed.). Cambridge University Press.

[6] Gries, D., & Schneider, F. B. (1993). *A Logical Approach to Discrete Math.* Springer.

[7] Thompson, S. (2011). *Haskell: The Craft of Functional Programming* (3rd ed.). Addison-Wesley.