Lógica computacional

Tema: Tableaux

Pilar Selene Linares Arévalo

Facultad de Ciencias
Universidad Nacional Autónoma de México

abril 2018

Material desarrollado bajo el proyecto UNAM-PAPIME PE102117.

Introducción

Los **tableaux** o *tablas semánticas* son un método de demostración por refutación: dado un conjunto de premisas Γ y una conclusión φ, para mostrar que $\Gamma \models \varphi$ se debe demostrar que el conjunto $\Gamma \cup \{\neg\varphi\}$ no tiene un modelo.

Este método se basa en la semántica más que en la sintáxis de las fórmulas. Sin embargo, se requiere de una clasificación de las fórmulas basada en sintaxis.

Esquemas

Aunque el método fue introducido en los años 1950 de manera independiente por Hintikka y Beth, fue Smullyan quien estandariza la notación y realiza la clasificación de las fórmulas en literales y cuatro tipos identificando su esquema:

Literal

Una **literal** es una fórmula atómica o la negación de una fórmula atómica.

Esquemas

El conjunto de fórmulas que no son literales se clasifica en cuatro tipos, en cada tipo de fórmula se distinguen ciertas subfórmulas necesarias posteriormente:

Tipo α

Tipo $\boldsymbol{\alpha}$: si la fórmula es conjuntiva o es equivalente a una fórmula conjuntiva.
Una fórmula $\chi \in$ FORM es tipo α, si tiene alguna de las siguientes formas:

1. $\chi = \varphi \wedge \psi$, con subfórmulas $\alpha_1 = \varphi$ y $\alpha_2 = \psi$.
2. $\chi = \neg(\varphi \vee \psi)$, con subfórmulas $\alpha_1 = \neg\varphi$ y $\alpha_2 = \neg\psi$.
3. $\chi = \neg(\varphi \to \psi)$, con subfórmulas $\alpha_1 = \varphi$ y $\alpha_2 = \neg\psi$.

Esquemas

Tipo β

Tipo β: si la fórmula es disyuntiva o es equivalente a una fórmula disyuntiva.

Una fórmula $\chi \in$ FORM es tipo β, si tiene alguna de las siguientes formas:

1. $\chi = \varphi \vee \psi$, con subfórmulas $\beta_1 = \varphi$ y $\beta_2 = \psi$.
2. $\chi = \neg(\varphi \wedge \psi)$, con subfórmulas $\beta_1 = \neg\varphi$ y $\beta_2 = \neg\psi$.
3. $\chi = \varphi \rightarrow \psi$, con subfórmulas $\beta_1 = \neg\varphi$ y $\beta_2 = \psi$.
4. $\chi = \varphi \leftrightarrow \psi$, con subfórmulas $\beta_1 = \varphi \wedge \psi$ y $\beta_2 = \neg\varphi \wedge \neg\psi$.
5. $\chi = \neg(\varphi \leftrightarrow \psi)$, con subfórmulas $\beta_1 = \neg\varphi \wedge \psi$ y $\beta_2 = \varphi \wedge \neg\psi$.

Esquemas

Tipo γ

Tipo $\boldsymbol{\gamma}$: si la fórmula está cuantificada universalmente o es equivalente a una fórmula cuantificada universalmente.

Una fórmula φ es de tipo γ, si tiene alguna de las siguientes formas:

1. $\chi = \forall x \varphi$.
2. $\chi = \neg \exists x \varphi$.

Tipo δ

Tipo $\boldsymbol{\delta}$: si la fórmula está cuantificada existencialmente o es equivalente a una fórmula cuantificada existencialmente.

Una fórmula φ es tipo γ, si tiene alguna de las siguientes formas:

1. $\chi = \exists x \varphi$.
2. $\chi = \neg \forall x \varphi$.

Tableaux

Un **tableau** es un árbol donde cada nodo está etiquetado con una fórmula y tiene a lo más dos hijos.

Inicialmente el árbol únicamente contiene una rama en donde cada uno de los nodos que la conforman están etiquetados con una de las fórmulas del conjunto $\Gamma \cup \{\neg\varphi\}$, si lo que queremos probar es que $\Gamma \models \varphi$; o bien, Γ si lo que queremos probar es que este conjunto es insatisfacible.

La construcción del árbol la dictan las siguientes reglas de extensión:

Tableau

Regla de extensión α:

$$\begin{array}{c} \varphi \wedge \psi \\ | \\ \varphi \\ | \\ \psi \end{array}$$

Regla de extensión β:

$$\begin{array}{ccc} & \varphi \vee \psi & \\ / & & \backslash \\ \varphi & & \psi \end{array}$$

Tableau

Regla de extensión γ:

$$\begin{array}{c} \forall x\varphi \\ | \\ \varphi[x := t] \end{array}$$

donde t es cualquier término cerrado. Las fórmulas tipo γ siempre están activas.

Tableau

Regla de extensión δ:

$$\begin{array}{c} \exists x\varphi \\ | \\ \varphi[x := c] \end{array}$$

donde c es una constante nueva, es decir una constante que no figura antes en la rama.

Tableaux

Ejemplos de reglas de extensión aplicadas a diferentes fórmulas:

- Extensión de fórmulas equivalentes tipo α:

$$\begin{array}{c} \neg(\varphi \vee \psi) \\ | \\ \neg\varphi \\ | \\ \neg\psi \end{array} \qquad\qquad \begin{array}{c} \neg(\varphi \to \psi) \\ | \\ \varphi \\ | \\ \neg\psi \end{array}$$

Tableaux

Ejemplos de reglas de extensión aplicadas a diferentes fórmulas:

- Extensión de fórmulas equivalentes tipo β:

$$\begin{tikzcd}[column sep=small]
 & \neg(\varphi \wedge \psi) \arrow[dl, dash] \arrow[dr, dash] & \\
\neg\varphi & & \neg\psi
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & \varphi \to \psi \arrow[dl, dash] \arrow[dr, dash] & \\
\neg\varphi & & \psi
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & \varphi \leftrightarrow \psi \arrow[dl, dash] \arrow[dr, dash] & \\
\varphi \wedge \psi & & \neg\varphi \wedge \neg\psi
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=small]
 & \neg(\varphi \leftrightarrow \psi) \arrow[dl, dash] \arrow[dr, dash] & \\
\neg\varphi \wedge \psi & & \varphi \wedge \neg\psi
\end{tikzcd}$$

Tableaux

Ejemplos de reglas de extensión aplicadas a diferentes fórmulas:

- Extensión de fórmulas equivalentes tipo γ:

$$\begin{array}{c} \neg\exists x\varphi \\ | \\ \neg\varphi[x := t] \end{array}$$

- Extensión de fórmulas equivalentes tipo δ:

$$\begin{array}{c} \neg\forall x\varphi \\ | \\ \neg\varphi[x := a] \end{array}$$

Tableaux

Tableaux semánticos

Sea $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ un conjunto de fórmulas, un **tableau** para Γ denotado $\mathcal{T}(\Gamma)$ es una extensión obtenida recursivamente como sigue:

1. El árbol formado por la rama cuyos nodos son $\varphi_1, \varphi_2, \ldots, \varphi_n$ es un tableau para Γ.
2. Si $\mathcal{T}_1$ es un tableau para Γ, entonces la extensión que se obtiene después de aplicarle alguna de las reglas α, β, γ, δ es un tableau para Γ.
3. Son todos

Tableaux

Tableau cerrado

Una rama ρ de un tableau es *cerrada* si tanto una literal (fórmula atómica) φ como su negación figuran en ρ.
Un tableau es **cerrado** si todas sus ramas están cerradas.

Tableau abierto

Una rama ρ de una tableau es *abierta* si no es cerrada. Un tableau se considera **abierto** si contiene al menos una rama abierta.

Tableaux

Correctud

Si hay un tableau cerrado para Γ entonces Γ no tiene un modelo.
Es decir, si un tableaux con conjunto inicial Γ se cierra entonces dicho conjunto es insatisfacible.

Completud refutacional

Si Γ no tiene un modelo entonces existe un tableau $\mathcal{T}(\Gamma)$ cerrado.
Es decir, si un conjunto inicial Γ es insatisfacible entonces existe una extensión $\mathcal{T}(\Gamma)$ cerrada.

Tableaux

Corolario 1
$\Gamma \models \varphi$ si y sólo si existe una extensión $\mathcal{T}(\Gamma \cup \{\neg\varphi\})$ cerrada.

Corolario 2
$\models \varphi$ si y sólo si existe una extensión $\mathcal{T}(\neg\varphi)$ cerrada.

Indecibilidad

Sabemos que la lógica de predicados es indecidible, es decir, *no existe un algoritmo que reciba una fórmula φ y decida si ésta es o no una fórmula universalmente válida*. En particular:

- No existe un algoritmo que reciba un conjunto de premisas Γ y una fórmula φ y decida si se cumple o no $\Gamma \models \varphi$.
- Podemos preguntarnos entonces por qué el método de tableaux no nos permitirá decidir la consecuencia lógica en la lógica de predicados.

Indecibilidad

Consideremos el siguiente argumento:

$$\forall x \exists y P(x, y),\ R(a, b)\ /\ \therefore \exists x R(x, b)$$

es claramente correcto pues la conclusión es consecuencia de la segunda premisa, la primera no aporta información.

Aunque existe un tableau cerrado (utilizando la segunda premisa y la negación de la conclusión) también existe un tableau infinito (utilizando la primera premisa).

Por lo tanto si en el análisis de un argumento nos encontramos ante un tableau que no se ha cerrado no podemos asegurar algo.

Tableaux

Los tableaux en lógica de predicados únicamente pueden ser utilizados para trata de establecer *insatisfacibilidad*, *validez* ó *consecuencia lógica* y por lo general **no** son un método apto para construir modelos como en el caso de la lógica de proposiciones.