Tableaux Semánticos
Lógica Computacional 2018-2, Nota de clase 8

Favio Ezequiel Miranda Perea Araceli Liliana Reyes Cabello
Lourdes Del Carmen González Huesca Pilar Selene Linares Arévalo

12 de abril de 2018
Facultad de Ciencias UNAM

1. Introducción

Los *tableaux* o *tablas semánticas* son un método de demostración por refutación, el cual fue introducido en los años 1950 de manera independiente por Hintikka y Beth. 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.

Esta teoría de prueba utiliza el principio de refutación: dado un conjunto de premisas Γ y una conclusión φ, se debe mostrar que el conjunto $\Gamma \cup \{\neg\varphi\}$ no tiene un modelo [1], lo cual es equivalente a que $\Gamma \models \varphi$ es decir, a que φ sea consecuencia lógica de Γ.

Es Smullyan quien estandariza la notación y realiza la clasificación de las fórmulas en cuatro tipos identificando su esquema:

- *tipo* α: las fórmulas conjuntivas.
- *tipo* β: las fórmulas disyuntivas.
- *tipo* γ: las fórmulas universales.
- *tipo* δ: las fórmulas existenciales.

De acuerdo a su tipo, una fórmula genera una extensión en el tableau. Veamos ahora los preliminares para después detallar el método de prueba.

Definición 1 *Una literal es una fórmula atómica o la negación de una fórmula atómica.*

Definición 2 *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*: si la fórmula es conjuntiva o es equivalente a una fórmula conjuntiva.*
Una fórmula $\chi \in \mathsf{FORM}$ *es tipo* α*, si tiene alguna de las siguientes formas:*

a) $\chi = \varphi \wedge \psi$, *con subfórmulas* $\alpha_1 = \varphi$ *y* $\alpha_2 = \psi$.
b) $\chi = \neg(\varphi \vee \psi)$, *con subfórmulas* $\alpha_1 = \neg\varphi$ *y* $\alpha_2 = \neg\psi$.
c) $\chi = \neg(\varphi \to \psi)$, *con subfórmulas* $\alpha_1 = \varphi$ *y* $\alpha_2 = \neg\psi$.

[1] Recordemos que un modelo de un conjunto de fórmulas Γ, es una interpretación $\mathcal{I}$ tal que para cualquier estado de las variables σ se cumple $\mathcal{I}_\sigma(\Gamma) = 1$.

Tipo *β: si la fórmula es disyuntiva o es equivalente a una fórmula disyuntiva.*
Una fórmula $\chi \in \mathsf{FORM}$ es tipo β, si tiene alguna de las siguientes formas:

a) $\chi = \varphi \vee \psi$, *con subfórmulas* $\beta_1 = \varphi$ *y* $\beta_2 = \psi$.

b) $\chi = \neg(\varphi \wedge \psi)$, *con subfórmulas* $\beta_1 = \neg\varphi$ *y* $\beta_2 = \neg\psi$.

c) $\chi = \varphi \rightarrow \psi$, *con subfórmulas* $\beta_1 = \neg\varphi$ *y* $\beta_2 = \psi$.

d) $\chi = \varphi \leftrightarrow \psi$, *con subfórmulas* $\beta_1 = \varphi \wedge \psi$ *y* $\beta_2 = \neg\varphi \wedge \neg\psi$.

e) $\chi = \neg(\varphi \leftrightarrow \psi)$, *con subfórmulas* $\beta_1 = \neg\varphi \wedge \psi$ *y* $\beta_2 = \varphi \wedge \neg\psi$.

Tipo *γ: 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:

a) $\chi = \forall x \varphi$.

b) $\chi = \neg\exists x \varphi$.

Tipo *δ: 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:

a) $\chi = \exists x \varphi$.

b) $\chi = \neg\forall x \varphi$.

De esta manera cualquier fórmula pertenece a una de cinco categorías: literales, α, β, γ ó δ.
Obsérvese que algunas equivalencias en fórmulas tipo β o sus negaciones pueden clasificarse como fórmulas α, pero la clasificación elegida aquí es óptima para construir los tableaux, en particular, la negación de una fórmula β no es necesariamente una fórmula α.

2. Tableaux Semánticos

Un tableau es un árbol donde cada nodo está etiquetado con una fórmula y tiene a lo más dos hijos. La construcción del árbol la dictan las reglas de extensión que introducimos enseguida. 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.

Definición 3 *Las reglas de extensión de tableaux se definen de acuerdo a la clasificación antes mencionada, y son:*

1. *Regla de extensión α:*

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

2. Regla de extensión β:

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

3. Regla de extensión γ:

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

donde t es cualquier término cerrado. Decimos que las fórmulas tipo γ siempre están activas.

4. 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.

Como se puede apreciar, cada extensión de la regla agreaga a lo más dos nodos con las subfórmulas correspondientes.

A pesar de que se consideran algunas fórmulas equivalentes en la clasificación, se debe enfatizar que su uso respeta las reglas de extensión y no debe procederse a una transformación de una fórmula en un principio. Veamos los casos de las reglas aplicadas a las fórmulas equivalentes:

1. Extensión de fórmulas equivalentes tipo α:

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

2. Extensión de fórmulas equivalentes tipo β:

$$\begin{array}{ccc} & \neg(\varphi \wedge \psi) & \\ / & & \backslash \\ \neg\varphi & & \neg\psi \end{array} \quad \begin{array}{ccc} & \varphi \to \psi & \\ / & & \backslash \\ \neg\varphi & & \psi \end{array} \quad \begin{array}{ccc} & \varphi \leftrightarrow \psi & \\ / & & \backslash \\ \varphi \wedge \psi & & \neg\varphi \wedge \neg\psi \end{array} \quad \begin{array}{ccc} & \neg(\varphi \leftrightarrow \psi) & \\ / & & \backslash \\ \neg\varphi \wedge \psi & & \varphi \wedge \neg\psi \end{array}$$

3. Extensión de fórmulas equivalentes tipo γ:

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

4. Extensión de fórmulas equivalentes tipo δ:

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

A continuación definimos formalmente un tableau semántico:

Definición 4 (*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*

Dado un conjunto inicial Γ denotamos con $\mathcal{T}(\Gamma)$ a cualquier extensión de Γ, obtenida mediante las reglas arriba definidas, la cual por supuesto no es única.
Las siguientes observaciones acerca del uso de las reglas de extensión son de gran importancia:

- Las reglas de extensión α, β, δ se utilizan una **única** vez para cada fórmula. Es decir, si φ es una fórmula a la que se puede aplicar alguna extensión α, β, δ entonces una vez hecha la extensión la fórmula φ no puede usarse nuevamente para una extensión.
- En contraste la regla γ puede utilizarse para realizar una extensión cualquier número de veces, por supuesto utilizando distintos términos cerrados t en cada ocasión. Por esto decimos que la regla γ permanece siempre activa.
- Cualquier extensión de una fórmula φ debe realizarse sobre **todas** las ramas del tableau a las que pertenece la fórmula φ y únicamente sobre éstas.
- Para extender un tableau es conveniente preferir el uso de fórmulas tipo α sobre fórmulas tipo β para buscando ramificar el tableau lo menos posible.
- Similarmente el uso de la regla γ se debe posponer hasta que sea estrictamente necesario.
- La regla γ debe utilizar en su extensión términos cerrados ya presentes en el tableau, excepto cuando estos aún no existan. Únicamente en este caso se permite aplicar la regla γ usando una nueva constante a.

Definición 5 (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.*

Definición 6 (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.*

2.1. Los teoremas de correctud y completud refutacional

El método de tableaux se relaciona con la semántica de la lógica mediante los siguientes teoremas de correctud y completud refutacional.

Teorema 1 (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.

Teorema 2 (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.

Con respecto a la consecuencia lógica se tienen los siguientes corolarios:

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.*

3. Tableaux en lógica de predicados: indecidibilidad

Es bien sabido 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 los algoritmos de decisión para la lógica de proposiciones fallan al usarlos o trasladarlos a la lógica de predicados, en particular por qué el método de tableaux no nos permitirá decidir la consecuencia lógica en la lógica de predicados. ¿Donde está la falla?

Considérese el siguiente tableau:

$$
\begin{array}{c}
\forall x \exists y P(x,y) \\
| \\
\exists y P(c_1, y) \\
| \\
P(c_1, c_2) \\
| \\
\exists y P(c_2, y) \\
| \\
P(c_2, c_3) \\
| \\
\exists y P(c_3, y) \\
| \\
P(c_3, c_4) \\
| \\
\vdots
\end{array}
$$

- Al aplicar la regla γ a la primera fórmula hacemos la extensión con una nueva constante c_1 obteniendo una fórmula existencial $\exists y P(c_1, y)$.
- Al aplicar la regla δ a $\exists y P(c_1, y)$ debemos elegir una constante c_2 distinta de c_1 obteniendo $P(c_1, c_2)$.
- Como las fórmulas universales están siempre activas entonces podemos aplicar la regla γ nuevamente a la primera fórmula pero ahora con la constante c_2 obteniendo $\exists y P(c_2, y)$, etc.
- Se observa que este proceso se puede continuar indefinidamente por lo que podemos obtener una rama infinita.
- Al existir ramas infinitas no tiene sentido hablar de ramas completas como en la lógica de proposiciones. Pueden existir ramas, como la de arriba que sigan extendiéndose indefinidamente, por lo que nunca sabremos si la rama se podrá cerrar o no.

De este ejemplo debe quedar claro el porque 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.

Existen formas de construir modelos a partir de un tableaux en lógica de predicados pero son más bien casos particulares basados en ciertas *heurísticas.*

En lógica de proposiciones la construcción de un modelo se sirve del concepto de rama completa y abierta, pero en lógica de predicados no existe un concepto de rama completa. Esto debe convencernos de la indecidibilidad de la lógica de predicados.

Veamos otro ejemplo, considérese lo siguiente:

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

este argumento es claramente correcto pues la conclusión es consecuencia de la segunda premisa, la primera no aporta nada pues habla de otra relación.
Aunque existe un tableau cerrado generado por el argumento (utilizando la segunda premisa y la negación de la conclusión) también existe un tableau infinito muy semejante al que analizamos en el ejemplo anterior (utilizando la primer premisa).
Por lo tanto si en el análisis de un argumento nos encontramos ante un tableau que no se ha cerrado no podemos asegurar nada.