Cálculo Proposicional y Cálculo de Predicados

Paula Severi

University of Leicester

Curso Noviembre-Diciembre 2018. Facultad de Ingeniería.
Universidad de la República.
Montevideo, Uruguay.

- Letras proposicionales. Suponemos tenemos un conjunto infinito de letras proposicionales que las denotamos como $p, q, r, \ldots$

Lógica proposicional. Sintaxis.

- Letras proposicionales. Suponemos tenemos un conjunto infinito de letras proposicionales que las denotamos como $p, q, r, \ldots$

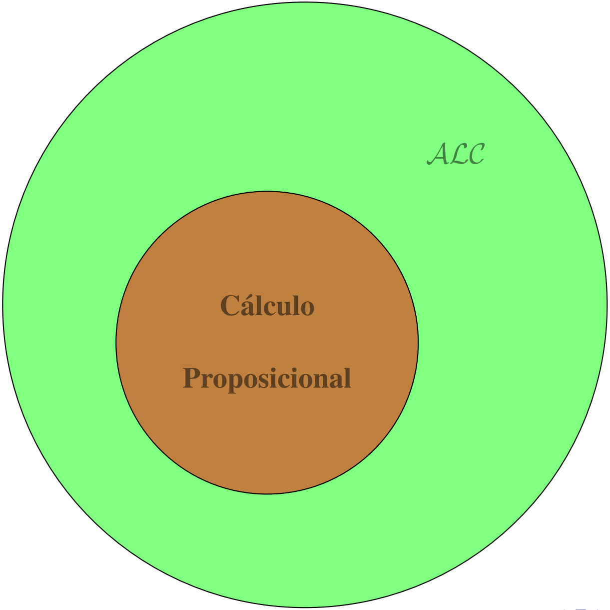
Fórmulas

Se denotan φ, ψ, etc y se definen inductivamente por medio de la gramática siguiente:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

donde p es una letra proposicional.

Cálculo proposicional vs $\mathcal{ALC}$



Tablas de Verdad

φ	$\neg\varphi$
True	*False*
False	*True*

φ	ψ	$\varphi \to \psi$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

φ	ψ	$\varphi \vee \psi$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

φ	ψ	$\varphi \wedge \psi$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*False*

Interpretación de una fórmula proposicional

Interpretación

Es una función $\mathcal{I}$ de las letras proposicionales en $\{True, False\}$. Esta interpretación se extiende a una fórmula arbitraria usando las tablas de verdad:

$\mathcal{I}(\varphi)$	$\neg\mathcal{I}(\varphi)$
True	*False*
False	*True*

$\mathcal{I}(\varphi)$	$\mathcal{I}(\psi)$	$\mathcal{I}(\varphi \to \psi)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

$\mathcal{I}(\varphi)$	$\mathcal{I}(\psi)$	$\mathcal{I}(\varphi \vee \psi)$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

$\mathcal{I}(\varphi)$	$\mathcal{I}(\psi)$	$\mathcal{I}(\varphi \wedge \psi)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*False*

Modelo y Satisfactibilidad

Modelo

$\mathcal{I}$ es modelo de una fórmula P si $\mathcal{I}(P) = \mathit{True}$.

Consistencia

Una fórmula (o conjunto de fórmulas) es consistente (satisfactible) si tiene modelo.

P es una tautología si y solo si $\neg P$ es insatisfactible.

Ejemplo de modelo de una fórmula

Ejemplo

	p	q	$p \vee \neg q$
$\mathcal{I}_1$	*True*	*True*	*True*
$\mathcal{I}_2$	*True*	*False*	*True*
$\mathcal{I}_3$	*False*	*True*	*False*
$\mathcal{I}_4$	*False*	*False*	*True*

Satisfactible

Esta fórmula tiene 3 modelos (la primera, la segunda y la cuarta filas).

Tenemos que $\mathcal{I}_1$ es modelo de $p \vee \neg q$ ya que:

$$\mathcal{I}_1(p \vee \neg q) = \mathit{True}$$

Ejemplo

p	q	$(p \vee q) \wedge \neg p \wedge \neg q$
True	*True*	*False*
True	*False*	*False*
False	*True*	*False*
False	*False*	*False*

Insatisfactible (contradicción)

Esta fórmula no tiene modelo.

Algoritmo para chequear satisfactibilidad basado en tablas de verdad

Ejemplo

p	q	r	$p \vee q$	$(p \vee q) \wedge r$
T	T	T	T	T
T	T	F	T	F
T	F	T	T	T
T	F	F	T	F
F	T	T	T	T
F	T	F	T	F
F	F	T	F	F
F	F	F	F	F

Complejidad del tiempo es exponencial

tenemos 2^n filas

Complejidad del espacio es polinomial

Para saber si es satisfactible alcanza con guardar una fila por vez. Cada fila tiene largo n.

Propositional Tableaux

Inventado por Beth (1955) y simplificado por Smullyan (1968).

Reglas de Tableaux

$$
\begin{tikzcd}[column sep=small]
 & \varphi \lor \psi \arrow[dl] \arrow[dr] & \\
\varphi & & \psi
\end{tikzcd}
\qquad
\begin{tikzcd}
\varphi \land \psi \arrow[d] \\
\varphi \\
\psi
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & \varphi \to \psi \arrow[dl] \arrow[dr] & \\
\neg\varphi & & \psi
\end{tikzcd}
$$

Propositional Tableaux

Inventado por Beth (1955) y simplificado por Smullyan (1968).

Reglas de Tableaux

$$
\begin{tikzcd}[column sep=small]
 & \varphi \lor \psi \arrow[dl] \arrow[dr] & \\
\varphi & & \psi
\end{tikzcd}
\qquad
\begin{tikzcd}
\varphi \land \psi \arrow[d] \\
\varphi \\
\psi
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & \varphi \to \psi \arrow[dl] \arrow[dr] & \\
\neg\varphi & & \psi
\end{tikzcd}
$$

Por simplicidad, en vez de agregar reglas para todas las negaciones aplicamos las siguientes equivalencias:

$$\neg\neg\varphi \leftrightarrow \varphi \qquad \neg(\varphi \lor \psi) \leftrightarrow \neg\varphi \land \neg\psi$$

$$\neg(\varphi \land \psi) \leftrightarrow \neg\varphi \lor \neg\psi \qquad \neg(\varphi \to \psi) \leftrightarrow \varphi \lor \neg\psi$$

$$(p \vee q) \wedge \neg p \wedge \neg q$$

Rama cerrada

Si una rama contiene una contradicción, esta rama se dice que está *cerrada*.

Insatisfactibilidad

Si todas las ramas estan cerradas, la fórmula es insatisfactible.

Tableaux para el cálculo proposicional

$$
\begin{array}{c}
(p \vee q) \wedge \neg p \wedge \neg q \\
\downarrow \\
(p \vee q) \quad \neg p \quad \neg q
\end{array}
$$

Rama cerrada

Si una rama contiene una contradicción, esta rama se dice que está *cerrada*.

Insatisfactibilidad

Si todas las ramas estan cerradas, la fórmula es insatisfactible.

Tableaux para el cálculo proposicional

$$(p \vee q) \wedge \neg p \wedge \neg q$$

$$\begin{tikzcd}
(p \vee q) \arrow[d] & \neg p \arrow[dr] & \neg q \\
p & & q
\end{tikzcd}$$

Rama cerrada

Si una rama contiene una contradicción, esta rama se dice que está *cerrada*.

Insatisfactibilidad

Si todas las ramas estan cerradas, la fórmula es insatisfactible.

Tableaux para el cálculo proposicional

$$(p \vee q) \wedge \neg p \wedge \neg q$$

$$(p \vee q) \qquad \neg p \qquad \neg q$$

$$p \qquad\qquad\qquad q$$

$$\checkmark \qquad\qquad\qquad \checkmark$$

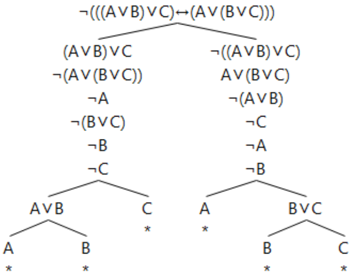
Rama cerrada

Si una rama contiene una contradicción, esta rama se dice que está *cerrada*.

Insatisfactibilidad

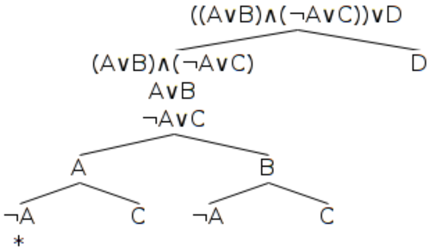
Si todas las ramas estan cerradas, la fórmula es insatisfactible.

Tableaux para el cálculo proposicional



Esta fórmula es insatisfactible ya que todas las ramas estan cerradas.

Tableaux para el cálculo proposicional



Tiene 4 modelos, *uno por cada rama abierta*, e.g. la interpretación que asigna $\mathcal{I}(D) = \mathit{True}$ y al resto cualquier cosa.

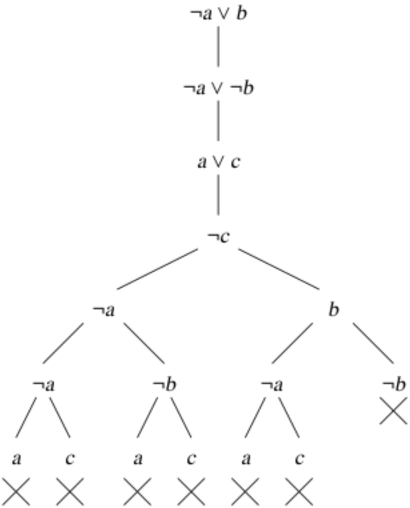
Tableaux Completo

Cuando no se pueden aplicar mas reglas.

Satisfactibilidad

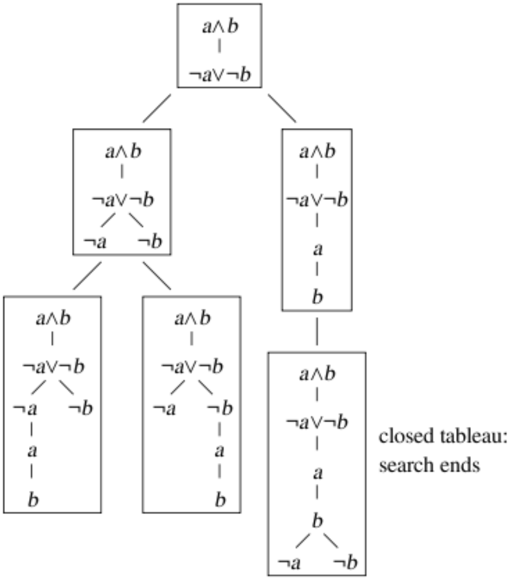
Si el tableaux es completo y hay una rama abierta entonces la fórmula es satisfactible.

Tableaux para el cálculo proposicional

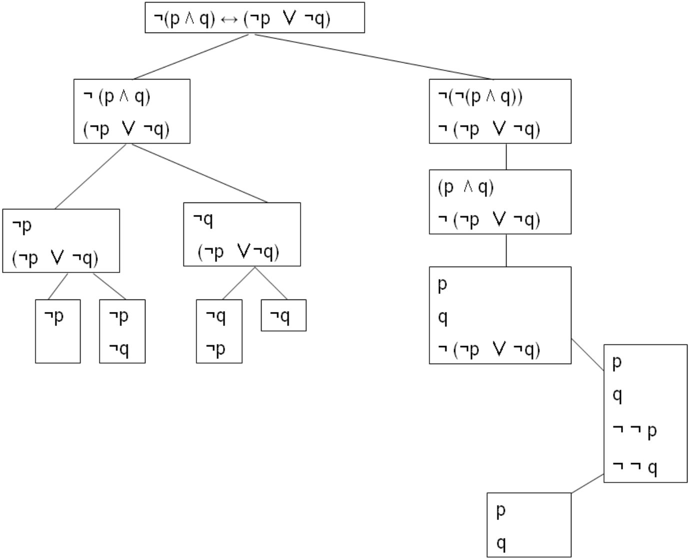


Esta fórmula es insatisfactible ya que todas las ramas estan cerradas.

Tableaux para el cálculo proposicional

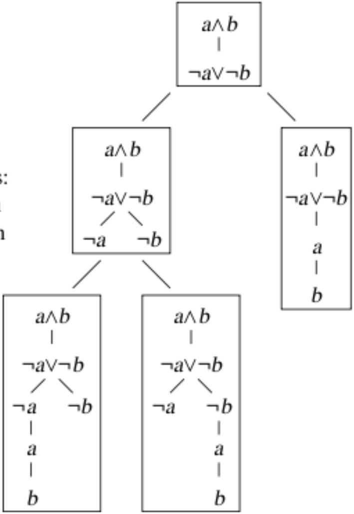


Tableaux para el cálculo proposicional



Tableaux para el cálculo proposicional

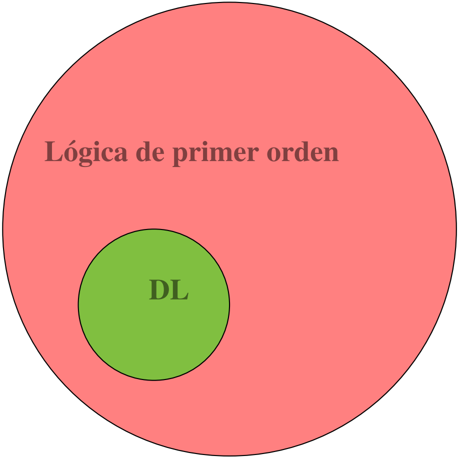
two choices:
expand ∧ on left branch
or on right branch



Cálculo de Predicados (Lógica de primer orden)

La lógica de primer orden tiene el poder expresivo suficiente para definir a prácticamente todas las matemáticas.

Mucho mas poder expresivo que cualquier DL.

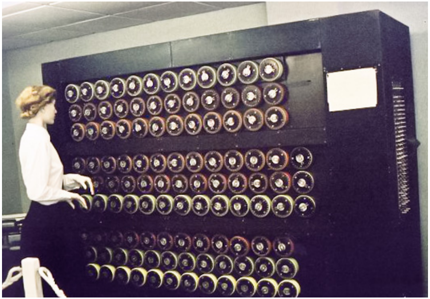
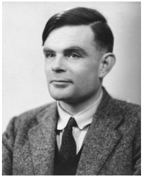


¿ Porqué no usamos la lógica de primer orden en vez de DL?

¿ Porqué no usamos la lógica de primer orden en vez de DL?

Teorema (Alonzo Church, 1936. Alan Turing 1937)

El problema de saber si una fórmula de la lógica de primer orden se deduce de los axiomas o no es indecidible.

Idea de la demostración: se reduce el problema de la parada al Entscheidungsproblem.

Términos

$$t := x \mid f(t_1, \dots, t_n)$$

donde f es una constante n-aria (n puede ser 0).

Términos

$$t := x \mid f(t_1, \ldots, t_n)$$

donde f es una constante n-aria (n puede ser 0).

Fórmulas

Se denotan φ, ψ, *etc* y se definen inductivamente por medio de la gramática siguiente:

$$\varphi := P(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \forall x.\varphi \mid \exists x.\varphi$$

donde P es un predicado n-ario y $t_1, \ldots, t_n$ son términos.

- Interpretacion tiene un dominio Δ, un conjunto arbitrario.
- Términos se interpretan como elementos del dominio Δ.
- Predicados son relaciones n-arias sobre Δ.

$\exists x.\mathsf{Sentado}(\mathsf{Edel}, x, \mathsf{Regina})$

Hay alguien sentado entre Edel y Regina.

$\exists xyz.\mathsf{Member}(x, y) \wedge \mathsf{Member}(y, z)$

Existe un conjunto z que contiene un conjunto y que contiene un elemento x.

Traducción de $\mathcal{ALC}$ a la lógica de primer orden

- conceptos atómicos $\Leftrightarrow$ predicados unarios
- roles atómicos $\Leftrightarrow$ predicados binarios
- conceptos $\Leftrightarrow$ fórmulas con una variable libre

Traducción de conceptos a la Lógica de Predicados

$$
\begin{aligned}
\pi_x(A) &= A(x) \\
\pi_x(\top) &= A_0(x) \to A_0(x) \\
\pi_x(\neg C) &= \neg(\pi_x(C)) \\
\pi_x(C \sqcap D) &= \pi_x(C) \wedge \pi_x(D) \\
\pi_x(\exists R.C) &= \exists y.(R(x,y) \wedge \pi_y(C))
\end{aligned}
$$

Traducción de TBox y ABox

$$
\begin{aligned}
\pi(\mathcal{T}) &= \bigwedge_{C \sqsubseteq D} (\forall x.\pi_x(C) \to \pi_x(D)) \\
\pi(\mathcal{A}) &= \bigwedge_{a:C \in \mathcal{A}} \pi_x(C)[x/a] \wedge \bigwedge_{(a,b):R \in \mathcal{C}} R(a,b)
\end{aligned}
$$

Consecuencia de la traducción a Lógica de Predicados

Teorema

Los problemas de inferencia en $\mathcal{ALC}$ son todos decidibles.

Idea de la demostración

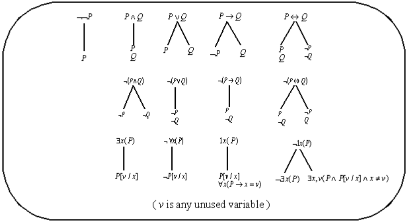
La traducción sólamente usa dos variables x, y. El fragmento de la lógica de predicados con dos variables es decidible.

Semantic Tableaux (after Wilfrid Hodges)

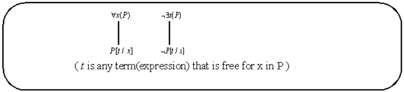
To show $P, Q \ldots, R \vdash S$ start with these formulae $P, Q \ldots, R, \neg S$

You can use Boolean algebra to change P to Q.
You can substitute equivalent propositions to convert P to Q.
* You can apply the equalities between expressions.
* You can use any axioms or algebraic laws that apply.

Apply one of these rules to all branches and you can check the root off .

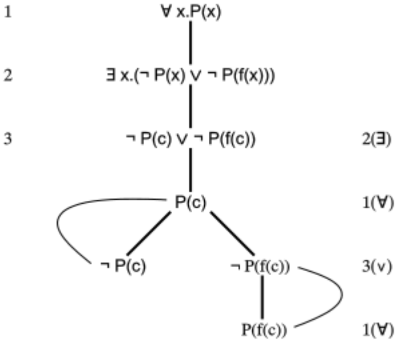


Apply one of these rules but don't check the root off.



Close any branch with both P and $\neg P$ on it. Or any branch with $t \neq t$. $\downarrow$ ✕

Tableaux para el cálculo de predicados



Bibliografía

- Hamilton, A.G. (1988), Logic for Mathematicians, Cambridge: Cambridge University Press.
- Ebbinghaus, H.-D.; Flum, J.; Thomas, W. (1994), Mathematical Logic, Nueva York: Springer.
- Hodges, Wilfrid (1977). Logic An Introduction to Elementary Logic. Penguin Books.
- Melvin Fitting. First-Order Logic and Automated Theorem Proving, Springer-Verlag, 1990.