Tablas Semánticas - Árboles de Refutación

- Método para determinar satisfacibilidad de fórmulas.

- Su uso se basa en la estrategia de refutación.

  - ✓ deducción por refutación

    $\models B$ sí y sólo sí $\neg B$ es contradicción

    $A \models B$ sí y sólo sí $\{A, \neg B\}$ es insatisfacible $(A = \{A_1, A_2, ..., A_n\})$

- Tabla semántica:

  Secuencia de fórmulas de la lógica de predicados de primer orden construidas de acuerdo a ciertas reglas, usualmente representada gráficamente como un árbol.

Árboles de Refutación

- Dadas A, B fórmulas de la lógica de predicados de primer orden:

Regla 1:	Regla 2:	Regla 3:	Regla 4:	
$\neg\neg A$	$A \wedge B$	$\neg(A \vee B)$	$\neg(A \rightarrow B)$	REGLAS de TIPO A
A	A	$\neg A$	A	
	B	$\neg B$	$\neg B$	

Regla 5:	Regla 6:	Regla 7:		
$A \vee B$	$\neg(A \wedge B)$	$A \rightarrow B$	premisas	REGLAS de TIPO B
A B	$\neg A$ $\neg B$	$\neg A$ B	conclusiones	

Árboles de Refutación

- **Reglas que involucran cuantificadores (todas son de tipo A):**

Regla 8:

$\forall x A(x)$

$A(t)$

t es un término cerrado

Regla 10:

$\exists x A(x)$

$A(c)$

c constante (nueva en el árbol)

Regla 9:

$\neg\exists x A(x)$

$\forall x \neg A(x)$

Regla 11:

$\neg\forall x A(x)$

$\exists x \neg A(x)$

Árboles de Refutación

- Las reglas de tipo ***A*** son las que tienen una o dos conclusiones en la misma rama.

Un modelo M satisface a la premisa sí y sólo sí M satisface a **todas** las conclusiones.

- Las reglas de tipo ***B*** son las que tienen dos conclusiones separadas en distintas ramas.

Un modelo M satisface a la premisa sí y solo sí M satisface **al menos una** de las conclusiones.

Heurística para aplicar reglas:
Aplicar las reglas de tipo A antes que las de tipo B.

Árboles de Refutación

Definiciones:

- Un **árbol de refutación** de una **fórmula F** se obtiene haciendo un número finito de **aplicaciones inmediatas de las reglas**, partiendo del **árbol** cuyo único nodo es **F**.

- Un **árbol de refutación** de un **conjunto finito C de fórmulas** es un á**rbol de refutación** de la **conjunción** de todas las **fórmulas de C.**

- Una **rama** de un árbol de fórmulas se dice **cerrada** si **contiene** a la vez una **fórmula y su negación**.
En **caso contrario**, se dice **abierta**.

- Un **árbol** se dice **cerrado** si **todas** sus **ramas** son **cerradas.**

Árboles de Refutación

- Un **árbol de refutación** se dice **completo** si cada una de sus ramas es **cerrada o saturada**.

(Una rama es saturada si no es cerrada y no se puede expandir aplicando las reglas)

- Toda fórmula F tiene por lo menos un **árbol de refutación completo**.

- Si F es una **fórmula satisfacible**, entonces **todo árbol de refutación de F** tiene **al menos una rama abierta**.

Árboles de Refutación

Sean H_1, H_2, …, H_n, F, C fórmulas de la lógica de predicados de primer orden:

- Si un conjunto finito $S = \{H_1, H_2, \ldots, H_n\}$, tiene un árbol de refutación cerrado, entonces S es insatisfacible.

- Si $\neg F$ tiene un árbol de refutación cerrado, entonces F es lógicamente válida.

- Si $\{H_1, H_2, \ldots, H_n, \neg C\}$ tiene un árbol de refutación cerrado, entonces $H_1, H_2, \ldots, H_n \models C$

Árboles de Refutación

Método para determinar si una sentencia A es válida:

1) Negar la fórmula A
2) Obtener un árbol de refutación para $\neg A$

✓ Si el árbol de refutación para $\neg A$ cerrado, $\neg A$ es insatisfacible → A es lógicamente válida es

✖ Si el árbol de refutación para $\neg A$ tiene al menos una rama abierta, $\neg A$ es satisfacible → A NO es lógicamente válida

Método para determinar si un razonamiento $H1, H2, \ldots, Hn \models C$ es válido:

Obtener un árbol de refutación para $H1 \wedge H2 \wedge \ldots \wedge Hn \wedge \neg C$

✓ Si el árbol de refutación es cerrado, $H1 \wedge H2 \wedge \ldots \wedge Hn \wedge \neg C$ es insatisfacible → $H1, H2, \ldots, Hn \models C$ es válido

✖ Si el árbol de refutación tiene al menos una rama abierta, $H1 \wedge H2 \wedge \ldots \wedge Hn \wedge \neg C$ es satisfacible → $H1, H2, \ldots, Hn \models C$ no es válido