Lógica Superior:

Computabilidad e Incompletitud

Profesor: Eduardo Barrio
UBA - Filosofía
2do cuatrimestre de 2015

Lógica y Algoritmos

- *¿Es posible construir una Máquina Universal?*

- *¿Es posible construir una máquina que resuelva todo problema lógico matemático?*

- *El tercer desafío de Hilbert: la decibilidad de la lógica de primer orden.*

Resultado de indecibilidad de la lógica de primer orden: la lógica formal no es mecánica.

La deducción no es totalmente mecanizable.

Recursividad: conjuntos y decibilidad

- Prueba de Gödel de la incompletitud de PA (1931)

Alonzo Church (1935–36) prueba la **indecibilidad de FOL usando funciones computables.**

- Es indecidible si dos expresiones del cálculo lamda son equivalentes.

Independientemente Alan Turing (1936–37) prueba la **indecibilidad de FOL usando máquinas de Turing.**

- El problema de la detención es irresoluble por una máquina. Y esté problema se reduce al problema de la indecibilidad.

PA es decidible ssi FOL es decidible (conjunción de Axiomas de PA → Teoremas de PA)

N no es decidible

Por lo tanto, FOL no es decidible

Decibilidad: verdad, validez y consecuencia

La FOL es completa: $\vDash$ **A sii** $\vdash$ **A** **(Gödel 1929)**

Si $\vDash$ A entonces ¬A no tiene modelo.

¿Es la verdad en M decidible**?**

M $\vDash$ **A**

Hay un algoritmo para computar si A es satisfacible o no.

¿Es la validez universal decidible:**?**

Γ $\vDash$ **A**

¿Es la relación de consecuencia lógica decidible:?

Hay un algoritmo para computar si todos los Modelos que satisfacen a **Γ**, también satisfacen a **A.**

Γ $\vdash$ **A** (dada completitud: si la implicación lógica fuera computable, también lo sería la noción de prueba)

Tres Problemas

¿Es verdad que hay un algoritmo que computa si una A de FOL es teorema?

¿Es verdad que hay un algoritmo que computa si una A de FOL es satisfacible o no?

¿Es verdad que si hubiera un algoritmo que computa si $\neg$A de FOL tiene modelo, entonces la noción de $\models$ A sería de decidible?

$\models$ A y M $\models \neg$A

Si $\not\models$A entonces **M** $\models$ $\neg$A

Si A no es universalmente válida en cualquier dominio de objetos, $\neg$A es satisfacible en estos dominios.

Si **M** $\models$ $\neg$A entonces $\not\models$ A

Si $\neg$A es satisfacible en M, A no es universalmente válida.

Problemas de Decisión

DETENCIÓN:

¿Hay un método efectivo que, aplicado a cualquier máquina de Turing M y cualquier input n, nos responde en un tiempo finito si M SE DETIENE a partir de n?

- Respuesta: NO (Asumiendo que *efectivo* significa *lo que una M puede hacer)*

Problemas de Decisión

DECISIÓN PARA IMPLICACIÓN LÓGICA:

¿Hay un método efectivo que, aplicado a cualquier conjunto Γ de fbf y cualquier fórmula A, nos responde en un tiempo finito si Γ IMPLICA A?

1. - Respuesta para la lógica proposicional: SI
2. - Respuesta para la lógica de predicados monádicos: SI
3. - Respuesta general para la lógica de predicados: NO

Estrategia de la prueba de la indecibilidad de FOL

Reducir el problema de la decisión de la lógica de primer orden al problema de la detención

Dada la generalidad de los lenguajes de primer orden, podemos representar en ellos el problema de la detención: representar inputs, estados, outputs, los actos de aplicar instrucciones para realizar una tarea para dar lugar a un paso ulterior de esa tarea.

De esta forma, determinar el problema de parada de una tarea efectiva (dado un input) se reduce a demostrar en Primer Orden la fórmula que expresa la existencia de un output a partir de la aplicación de una serie de instrucciones previamente fijadas.

Por tanto, si la Lógica de Primer Orden resulta ser decidible, el problema de la detención resultará igualmente decidible.

Si FOL es **decidible**, el problema de la detención es **mecánicamente soluble**.

El problema de la detención no es mecánicamente soluble.

Por lo tanto, FOL es indecidible.

Estrategia de la prueba de la indecibilidad de FOL

Reducir el problema de la decisión de la lógica de primer orden al problema de la detención

Si se puede resolver el problema de la decisión de LPO, se puede resolver el problema de la detención.

- Dado cualquier M y cualquier n, hay un Γ y un A tal que
  - **M para a partir del input n ↔ Γ implica D**

Por eso, si hubiera un método efectivo por medio del cual resolver mecánicamente en general si **Γ implica D,** habría un modo de resolver mecánicamente el problema de la detención. Pero, no lo hay! Por eso, el problema de la decisión de LPO es insoluble (mecánicamente).

Estrategia de la prueba de la indecibilidad de FOL

Sea T la conjunción de todas las fórmulas que describan todas las transiciones de M. Sea q_0 el estado inicial de M y q_a el estado de aceptación de M.

Sea A la fórmula que describe en FOL el input s, el estado inicial, las transiciones de M (T) junto con sus pasos t y el estado de aceptación (dado el input s).

La forma de A será un condicional: T & q_0 → ∃t ∃x (Q_it & Acepta tx & **$M_s$tx).**

Quisiéramos probar:

M acepta s sii A es válida

Lo cual reduce el problema de la validez al problema de la detención.

Pero, si FOL fuera decidible, para toda M sería posible determinar si PARA dado cualquier s.

La insolubilidad del problema de la **DETENCIÓN** muestra que FOL es indecidible.

Prueba de la indecibilidad de FOL

Técnica (recuerde el capítulo 8): representar M y la cinta usando Γ **y D**

1. El lenguaje e interpretación para las oraciones Γ y D

- Comience con una máquina **M** y un input **n.**
- M y n permanecen fijos.
- El lenguaje y la interpretación varía de acuerdo a M y n.

El lenguaje:

Constante: **0**
Predicados monádicos: $\mathbf{Q_1,\ldots,Q_k}$ [M es uns k-estado de máquina sin contar su estado STOP]

Predicados diádicos: **S, <**

Adicionales: **@, M**

Prueba de la indecibilidad de FOL

La *interpretación Estándar, M:*

(1) Dominio |M|: los enteros positivos y negativos. {0,+1,-1,+2,-2,...}

- Pasos en la computación estarán numerados por enteros no negativos, comenzando con 0.

- Las celdas sobre la cinta estarán numeradas por todos los enteros, con el puntero de partida en la celda 0.

Prueba de la indecibilidad de FOL

Denotaciones

- 0^M = 0

- $Q_iM(t)$ ↔ M está en el estado *i* en el tiempo *t.*
La denotación de **Q_i** es el conjunto de pasos en los cuales la máquina M está en el estado *i.*

El estado inicial es 1. Por eso, **$Q_1^M(0)$.**
Si M va al estado 2, entonces **$Q_2^M(1)$**

Si M eventualmente SE DETIENE, cada denotación será un conjunto de times.

- "S" *denota* la relación de sucesor: $S^M(m, n) \leftrightarrow n = m+1$

- "<" *denota* menor que.

- @ denota todos los pares (t,x) donde t es mayor o igual a 0 y M está en la celda x al tiempo t.

- M denota todos los pares (t, x) donde t es mayor o igual a 0 y la celda x está marcada con 1 en el tiempo t.

Cuando estamos hablando de @ y M siempre usaremos t para el primer argumento (tiempo o paso) y x para el segundo (celda).

Prueba de la indecibilidad de FOL

2.- El conjunto de oraciones Γ **Tres tipos**

(a) **Background acerca de S y** <

- único sucesor,
- predecesor,
- todo número es menor que su sucesor,
- la relación menor que es transitiva,
- ningún número es menor que sí mismo.

(b) **Descripción del tiempo 0**

Al comenzar, M está en el estado 1, en la celda 0 y sólo las celdas 0, 1, ..., n están marcadas con "1".

Por eso, hay una descripción de la configuración inicial usando @ y M:

$Q_1 0$ & @00 & M00 & M01 & ... & M0n & $\forall x((x \neq 0$ & $x \neq 1$& ... & $x \neq n) \rightarrow \neg M0x$

(c) **Descripción de M (una oración para cada instrucción de NO DETENCIÓN)**

Cada Instrucción tiene la forma:
Si en el estado *i* leyendo el símbolo *s*, entonces tome {una de cuatro acciones} y go to en el nuevo estado

(*s* puede ser 0 o 1)

Prueba de la indecibilidad de FOL

3. La oración D

Cada inmediata instrucción de **DETENCIÓN** es del tipo:

- Si en el estado i leyendo el símbolo s, entonces (haga algo) y go to (estado de parada).

Considérese la siguiente oración:

$$\exists t\ \exists x\ (Q_i t\ \&\ @tx\ \&\ \mathbf{M_s tx}).$$

Esta oración es verdadera ssi la máquina en algún punto (algún tiempo (paso) y celda) está en el estado *i* y leyendo el símbolo *s,* en el cual se para.

Por eso sea D la disyunción de todas las oraciones (todas las instrucciones de pre-halting). Entonces M se detiene ssi uno de los disyuntos es verdaderos i.e., si D es verdadero.

Prueba de la indecibilidad de FOL

4. **M para a partir del input n ↔ Γ implica D**

CASO 1: supongamos que M no se detiene dado el input *n. Entonces* D es falso en la interpretación estándar: ninguno de los disyuntos es verdaderos (de otra forma M pararía). Pero toda oración de Γ es verdadera. Por eso, existe una interpretación que hace a Γ verdadera y a D falsa; Por lo tanto, Γ no implica a D.

CASO 2: mostrar que si M se detiene, entonces Γ implica D.

La oración que describe para cada celda si está marcada o no.

Q_1a & @ap & Maq_1 & ... & Maq_n & $\forall x((x \neq q_1$ & ... & $x \neq q_n) \rightarrow \neg Max$

- Ahora, supongamos que M se detiene en un tiempo *b* = *a+1.*
- Entonces en a, la instrucción fue de una preparada.
  - Por eso, uno de los disyuntos de D es implicado por (vía generalización existencial para t y x):
    Q_ia & @ap & **$M_s$ap**
    donde **s** es el símbolo escrito en la celda p en el tiempo a. Pero esta oración es implicada por (*a) y Γ, y por lo tanto, D es implicada por (*a) y Γ.
- **Lemma:** Si *a≥0* y *b=a+1* es un tiempo en el cual la máquina aún no ha parado, entonces Γ más (*a) implica (*b).

Prueba de la indecibilidad de FOL

4. **M para a partir del input n ⇔ Γ implica D**

- **Lemma:** Si *a≥0* y *b=a+1* es un tiempo en el cual la máquina aún no ha parado, entonces Γ más (*a) implica (*b).
- Este lemma completa la prueba. (*0) es parte de Γ, y Γ más (*0) implica (*1), por eso Γ implica (*1). Continuando de esta forma, Γ implica (*2), ..., (*a) donde *b=a+1* es el tiempo en el cual M se detiene. Y como hemos visto que D es implicado por (*a) y Γ; por lo tanto D es implicado por Γ. Por lo tanto, Si M se detiene, D es implicado por Γ.