Lógica y Computabilidad

2do cuatrimestre 2020 - **A DISTANCIA**

Departamento de Computación - FCEyN - UBA

Lógica de Primer Orden - clase 4

Aplicaciones de compacidad, indecidibilidad de la lógica de primer orden

Aplicaciones de Compacidad - no expresividad

Teorema

Si Γ tiene modelos arbitrariamente grandes, tiene modelo infinito.

Demostración.

Definimos (en el lenguaje con solo la igualdad)

$$\begin{aligned}
\varphi_2 &= (\exists x)(\exists y) x \neq y \\
\varphi_3 &= (\exists x)(\exists y)(\exists z)(x \neq y \wedge x \neq z \wedge y \neq z) \\
&\vdots \\
\varphi_n &= \text{“hay al menos } n \text{ elementos”}
\end{aligned}$$

- por hipótesis, todo subconjunto finito de $\Gamma \cup \{\varphi_i \mid i \geq 2\}$ tiene modelo
- por Compacidad, $\Gamma \cup \{\varphi_i \mid i \geq 2\}$ tiene algún modelo $\mathcal{M}$
- $\mathcal{M}$ tiene que ser infinito $\square$

Conclusión:

- $\mathcal{A}$ es infinito sii $\mathcal{A} \models \{\varphi_i \mid i \geq 2\}$
- no existe Γ tal que $\mathcal{A}$ es finito sii $\mathcal{A} \models \Gamma$

Aplicaciones de Compacidad - modelos no estándar

Consideremos un lenguaje $\mathcal{L} = \{0, S, <, +, \cdot\}$ con igualdad. Consideremos la estructura $\mathcal{N} = \langle \mathbb{N}; 0, S, <, +, \cdot \rangle$ con la interpretación usual. Sea

$$\mathsf{Teo}(\mathcal{N}) = \{\varphi \in \mathsf{FORM}(\mathcal{L}) : \varphi \text{ es sentencia y } \mathcal{N} \models \varphi\}$$

Expandimos el lenguaje con una nueva constante c y definimos

$$\Gamma = \{0 < c, S(0) < c, S(S(0)) < c, S(S(S(0))) < c, \dots\}$$

- cada subconjunto finito de $\Gamma \cup \mathsf{Teo}(\mathcal{N})$ tiene modelo
- por Compacidad, $\Gamma \cup \mathsf{Teo}(\mathcal{N})$ tiene modelo
- por Löwenheim-Skolem $\Gamma \cup \mathsf{Teo}(\mathcal{N})$ un modelo numerable

$$\mathcal{M} = \langle M; 0^{\mathcal{M}}, S^{\mathcal{M}}, <^{\mathcal{M}}, +^{\mathcal{M}}, \cdot^{\mathcal{M}}, c^{\mathcal{M}} \rangle$$

- sea $\mathcal{M}'$ la restricción de $\mathcal{M}$ al lenguaje original $\mathcal{L}$
- $\mathcal{N} \models \varphi$ sii $\mathcal{M}' \models \varphi$ para toda sentencia $\varphi \in \mathsf{FORM}(\mathcal{L})$
  - $\mathcal{N} \models \varphi \quad \Rightarrow \quad \varphi \in \mathsf{Teo}(\mathcal{N}) \quad \Rightarrow \quad \mathcal{M} \models \varphi \quad \Rightarrow \quad \mathcal{M}' \models \varphi$
  - $\mathcal{N} \not\models \varphi \quad \Rightarrow \quad \mathcal{N} \models \neg\varphi \quad \Rightarrow \quad \neg\varphi \in \mathsf{Teo}(\mathcal{N}) \quad \Rightarrow \quad \mathcal{M} \models \neg\varphi \quad \Rightarrow \quad \mathcal{M}' \models \neg\varphi \quad \Rightarrow \quad \mathcal{M}' \not\models \varphi$
- $\mathcal{N}$ y $\mathcal{M}'$ no son isomorfos: $c^{\mathcal{M}}$ es inalcanzable en $\mathcal{M}'$

Repaso de Máquina Turing

Fijamos $\Sigma = \{1, *\}$.
Recordar que una máquina de Turing es una tupla

$$M = (\Sigma, Q, T, q_0, q_f)$$

donde

- Σ (finito) es el conjunto símbolos que puede escribir en la cinta
- Q (finito) es el conjunto de estados
  - tiene dos estados distinguidos:
    - $q_0 \in Q$ es el estado inicial
    - $q_f \in Q$ es el estado final
- $T \subseteq Q \times \Sigma \times \Sigma \cup \{L, R\} \times Q$ es la tabla finita de instrucciones

Modelo de cómputo de máquina de Turing

Recordar que la máquina

$$M = (\Sigma, Q, T, q_0, q_f)$$

con entrada $w \in \{1\}^+$ termina (notado $M(w)\downarrow$) sii partiendo de w en la cinta de entrada y la cabeza leyendo el primer caracter después de w,

$$\begin{array}{ccccccccc} \dots & * & * & 1 & 1 & \dots & 1 & * & * & \dots \\ & & & & & & & q_0 & & \end{array}$$

llega al estado q_f después de una cantidad finita de pasos.

No es computable determinar si una máquina de Turing termina o no.

Idea de la demostración de que Primer Orden es indecidible

- fijar un lenguaje adecuado $\mathcal{L}$
- dada una máquina M y $w \in \{1\}^+$, construir (uniformemente) una sentencia $\varphi_{M,w} \in \mathrm{FORM}(\mathcal{L})$ tal que
$$M(w)\downarrow \quad \text{sii} \quad \vdash \varphi_{M,w}$$
- si el problema de determinar si vale $\vdash \psi$ o $\nvdash \psi$ para $\psi \in \mathrm{FORM}(\mathcal{L})$ fuese computable, en particular sería computable determinar si $\vdash \varphi_{M,w}$ o $\nvdash \varphi_{M,w}$ para cualquier máquina M y entrada w.
- como esto último es no-computable, tampoco es computable determinar si vale $\vdash \psi$ o $\nvdash \psi$ para cualquier $\psi \in \mathrm{FORM}(\mathcal{L})$

Dados M y w, ¿quién es $\varphi_{M,w}$?

- una fórmula de $\mathcal{L}$ que se construye computablemente a partir de M y w
- $\varphi_{M,w}$ describe el comportamiento de M con entrada w en una cierta interpretación $\mathcal{A}$
- $\varphi_{M,w}$ es una fórmula-programa

El lenguaje $\mathcal{L}$

- símbolos de constante:
  - uno solo: ϵ
- símbolos de función:
  - la función 1 unaria
  - la función * unaria
- símbolos de relación:
  - infinitos (tantas como necesitemos) símbolos de relaciones binarias
  - sea $E = \{q_0, q_f, p, q, r, \dots\}$ un conjunto infinito de estados que podemos llegar a usar en máquinas de Turing
    - cada máquina particular usará solo una cantidad finita de estados de E
  - los símbolos des relación son:

$$R_{q_0}, R_{q_f}, R_p, R_q, R_r, \dots$$

Notación de los términos de $\mathcal{L}$

- si t es un término de $\mathcal{L}$, $1(t)$ lo notamos $1t$
- si t es un término de $\mathcal{L}$, $*(t)$ lo notamos $*t$

Por ejemplo

- $1(x)$ lo notamos $1x$
- $1(1(\epsilon))$ lo notamos 11ϵ
- $1(1(*(1(y))))$ lo notamos $11 * 1y$

La interpretación $\mathcal{A}$

Dada una máquina

$$M = (\Sigma, Q, T, q_0, q_f)$$

y una entrada $w \in \{1\}^+$, fijamos una interpretación $\mathcal{A} = \mathcal{A}_{M,w}$

- el universo: $A = \{1, *\}^* =$ cadenas finitas sobre $\{1, *\}$
  - va a representar datos en la cinta de M
- $\epsilon_{\mathcal{A}} =$ cadena vacía
  - la cinta es infinita, pero infinitos $*$ se representan como ϵ
- las funciones

  $$1_{\mathcal{A}} : A \to A \qquad \text{y} \qquad *_{\mathcal{A}} : A \to A$$

  se interpretan así:
  - $1_{\mathcal{A}}(x) = 1x$, o sea la cadena que empieza por 1 y sigue con x
  - $*_{\mathcal{A}}(x) = *x$, o sea la cadena que empieza por $*$ y sigue con x
- para $q \in Q$, $(R_q)_{\mathcal{A}}(x, y)$ es verdadero sii la máquina M con entrada w llega a una configuración en la que:
  - el estado es q
  - en la cinta está escrito x en orden inverso y a continuación y
  - la cabeza de M apunta al primer caracter de y

Definición de la fórmula-programa $\varphi_{M,w}$

Dada una máquina

$$M = (\Sigma, Q, T, q_0, q_f)$$

y una entrada

$$w = \overbrace{1 \ldots 1}^{k}$$

fijamos la interpretación $\mathcal{A} = \mathcal{A}_{M,w}$ que acabamos de ver.

- $\varphi_0 := R_{q_0}(\overbrace{1 \ldots 1}^{k} \epsilon, \epsilon)$
    - dice: "el estado inicial es alcanzable"
    - $\mathcal{A} \models \varphi_0$
- $\varphi_f := (\exists x)(\exists y)\ R_{q_f}(x, y)$
    - dice: "el estado final es alzanzable "
    - $\mathcal{A} \models \varphi_f$ sii $M(w) \downarrow$

Definición de la fórmula-programa $\varphi_{M,w}$

Para cada instrucción $I \in T$:

- si I dice *si M está en el estado q y lee un 1, escribir b y pasar al estado r*, definir

$$\psi_I \quad := \quad (\forall x)(\forall y)\ (R_q(x, 1y) \rightarrow R_r(x, by))$$

- si I dice *si M está en el estado q y lee un $*$, escribir b y pasar al estado r*, definir

$$\begin{aligned} \psi_I \quad := \quad & (\forall x)(\forall y)\ (R_q(x, *y) \rightarrow R_r(x, by)) \wedge \\ & (\forall x)\ (R_q(x, \epsilon) \rightarrow R_r(x, b\epsilon)) \end{aligned}$$

Definición de la fórmula-programa $\varphi_{M,w}$

- si I dice *si M está en el estado q y lee un 1, moverse a la izquierda y pasar al estado r*, definir

$$\begin{aligned}\psi_I \quad := \quad & (\forall x)(\forall y)\ (R_q(1x, 1y) \to R_r(x, 11y)) \wedge \\ & (\forall x)(\forall y)\ (R_q(*x, 1y) \to R_r(x, *1y)) \wedge \\ & (\forall y)\ (R_q(\epsilon, 1y) \to R_r(\epsilon, *1y))\end{aligned}$$

- si I dice *si M está en el estado q y lee un $*$ moverse a la izquierda y pasar al estado r*, definir

$$\begin{aligned}\psi_I \quad := \quad & (\forall x)(\forall y)\ (R_q(1x, *y) \to R_r(x, 1*y)) \wedge \\ & (\forall x)(\forall y)\ (R_q(*x, *y) \to R_r(x, **y)) \wedge \\ & (\forall y)\ (R_q(\epsilon, *y) \to R_r(\epsilon, **y)) \wedge \\ & (\forall x)\ (R_q(*x, \epsilon) \to R_r(x, \epsilon)) \wedge \\ & (\forall x)\ (R_q(1x, \epsilon) \to R_r(x, 1\epsilon)) \wedge \\ & (R_q(\epsilon, \epsilon) \to R_r(\epsilon, \epsilon))\end{aligned}$$

(Misma idea con *moverse a la derecha*...)

Definición de la fórmula-programa $\varphi_{M,w}$

Recordar que la máquina

$$M = (\Sigma, Q, T, q_0, q_f)$$

tiene siempre un conjunto finito de instrucciones T

Definimos

$$\varphi_{M,w} \quad := \quad (\varphi_0 \wedge \bigwedge_{I \in T} \psi_I) \to \varphi_f$$

Proposición

Si $\mathcal{A} \models \varphi_{M,w}$ *sii* $M(w)\downarrow$.

Demostración.

Sabemos que $\mathcal{A} \models \varphi_0$. Sabemos que $\mathcal{A} \models \varphi_f$ sii $M(w)\downarrow$. Es fácil ver que $\mathcal{A} \models \psi_I$ para cada $I \in T$.
Luego $\mathcal{A} \models \varphi_{M,w}$ sii $\mathcal{A} \models \varphi_f$ sii $M(w)\downarrow$. $\square$

Entscheidungsproblem

Teorema

$\vdash \varphi_{M,w}$ *sii* $M(w)\downarrow$.

Demostración.

$(\Rightarrow)$ Si $\vdash \varphi_{M,w}$ entonces $\models \varphi_{M,w}$, es decir, $\varphi_{M,w}$ es verdadera en toda interpretación. En particular, $\mathcal{A} \models \varphi_{M,w}$. Luego $M(w)\downarrow$.

$(\Leftarrow)$ Idea. Si $M(w)\downarrow$ entonces existe un cómputo de $M(w)$:

$$(x_1, r_1, y_1) \rightsquigarrow (x_2, r_2, y_2) \rightsquigarrow \cdots \rightsquigarrow (x_n, r_n, y_n)$$

con $x_i, y_i \in \{1, *\}^*, r_i \in Q$, $x_1 = w$, $r_1 = q_0$, $y_1 = \epsilon$, $r_n = q_f$.
Cada (x_i, r_i, y_i) representa una configuración del cómputo $M(w)$:

- el estado es r_i
- la cinta contiene $\cdots * * * [\quad x_i \quad][\quad y_i \quad] * * * \cdots$
- la cabeza está apuntando al primer caracter de y_i

Cada paso de la ejecución coincide con una sustitución de una de las fórmulas ψ_I.

- cómputo de $M(w)$ = demostración de $\varphi_{M,w}$
- fórmula $\varphi_{M,w}$ = programa de M

Programa = Demostración

Recordemos que

$$\varphi_{M,w} \quad := \quad (\varphi_0 \wedge \bigwedge_{I \in T} \psi_I) \to \varphi_f$$

Supongamos que M con entrada 111 da el siguiente cómputo:

$$(111, q_0, *) \overset{(q_0,*,L,q_1)\in T}{\rightsquigarrow} (11, q_1, 1) \overset{(q_1,1,L,q_f)\in T}{\rightsquigarrow} (1, q_f, 11)$$

Veamos que $\{\varphi_0\} \cup \{\psi_I \mid I \in T\} \vdash \varphi_f$. Esto prueba que $\vdash \varphi_{M,w}$.

- Recordemos que
  $\psi_{(q_0,*,L,q_1)} := \ldots \wedge (\forall x)\ (R_{q_0}(1x, \epsilon) \to R_{q_1}(x, 1\epsilon)) \wedge \ldots$
  Por SQ4, $\{\varphi_0\} \cup \{\psi_I \mid I \in T\} \vdash (R_{q_0}(111\epsilon, \epsilon) \to R_{q_1}(11\epsilon, 1\epsilon))$
- Recordemos que
  $\psi_{(q_1,1,L,q_f)} := (\forall x)(\forall y)\ (R_{q_1}(1x, 1y) \to R_{q_f}(x, 11y)) \wedge \ldots$
  Por SQ4, $\{\varphi_0\} \cup \{\psi_I \mid I \in T\} \vdash (R_{q_1}(11\epsilon, 1\epsilon) \to R_{q_f}(1\epsilon, 11\epsilon))$
- Recordemos que $\varphi_0 := R_{q_0}(111\epsilon, \epsilon)$
- Por MP, concluimos $\{\varphi_0\} \cup \{\psi_I \mid I \in T\} \vdash R_{q_f}(1\epsilon, 11\epsilon)$
- De esto, se puede concluir $\{\varphi_0\} \cup \{\psi_I \mid I \in T\} \vdash \underbrace{(\exists x)(\exists y)\ R_{q_f}(x, y)}_{\varphi_f}$

Entscheidungsproblem

Teorema (Turing, 1936)

Sea $\mathcal{L}$ el lenguaje descripto y sea $\psi \in \text{FORM}(\mathcal{L})$. El problema de decidir si $\vdash \psi$ o $\nvdash \psi$ no es computable.

Demostración.

Supongamos que hay un programa que dada $\psi \in \text{FORM}(\mathcal{L})$ devuelve *verdadero* sii $\vdash \psi$.

Dada M y w, habría un procedimiento para decidir si $M(w)\downarrow$ o $M(w)\uparrow$:

1. construir $\varphi_{M,w}$ (esto se hace computablemente)
2. si $\vdash \varphi_{M,w}$ entonces $M(w)\downarrow$; si no $M(w)\uparrow$

$\square$