Decidibilidad y definibilidad en lógica de primer orden

Semana $(9)_2 = 1001$

Lógica para Ciencia de la Computación - IIC2213

Prof. Sebastián Bugedo

Programa

Obertura

Primer acto
- Satisfacibilidad
- Decidibilidad

Intermedio

Segundo acto
- Definibilidad
- Isomorfismos

Epílogo

Programa

Vocabulario

Notación

Un **vocabulario** $\mathcal{L}$ es la unión de tres conjuntos dados por

$$\begin{aligned} \text{constantes} &: \{c_1, \ldots, c_\ell, \ldots\} \\ \text{funciones} &: \{f_1, \ldots, f_m, \ldots\} \\ \text{relaciones} &: \{R_1, \ldots, R_n, \ldots\} \end{aligned}$$

Dada una función f o relación R del vocabulario $\mathcal{L}$, diremos que su **aridad** es el número de argumentos que posee.

- f puede tener aridad mayor a 0
- R puede tener aridad mayor o igual a 0

Diremos que los elementos de $\mathcal{L}$ son los **símbolos del vocabulario**

Sintaxis de LPO: términos

Sea $\mathcal{L}$ un vocabulario. Supondremos además una lista infinita de variables conocida.

Definición

El conjunto de **$\mathcal{L}$-términos** es el menor conjunto que satisface

- Cada constante $c \in \mathcal{L}$ es un $\mathcal{L}$-término
- Cada variable x es un $\mathcal{L}$-término
- Si $t_1, \ldots, t_n$ son $\mathcal{L}$-términos y $f \in \mathcal{L}$ es una función n-aria, entonces $f(t_1, \ldots, t_n)$ es un $\mathcal{L}$-término

Sintaxis de LPO: fórmulas

Definición

El conjunto de **$\mathcal{L}$-fórmulas** es el menor conjunto que satisface

- Si t_1, t_2 son $\mathcal{L}$-términos, entonces $t_1 = t_2$ es $\mathcal{L}$-fórmula
- Si $t_1, \ldots, t_n$ son $\mathcal{L}$-términos y $R \in \mathcal{L}$ una relación n-aria, entonces $R(t_1, \ldots, t_n)$ es $\mathcal{L}$-fórmula
- Si φ, ψ son $\mathcal{L}$-fórmulas, entonces $(\neg\varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \to \psi)$ y $(\varphi \leftrightarrow \psi)$ son $\mathcal{L}$-fórmulas
- Si φ es $\mathcal{L}$-fórmula que menciona la variable x, entonces $(\exists x.\ \varphi)$ y $(\forall x.\ \varphi)$ son $\mathcal{L}$-fórmulas

Diremos que las $\mathcal{L}$-fórmulas de la forma
$t_1 = t_2$ y $R(t_1, \ldots, t_n)$ son **fórmulas atómicas**

Semántica de LPO: estructuras

Definición

Una **$\mathcal{L}$-estructura** $\mathfrak{A}$ es una tupla que contiene

- Un dominio $A \neq \varnothing$
- Para cada símbolo de constante $c \in \mathcal{L}$, se tiene un elemento
$$c^{\mathfrak{A}} \in A$$
- Para cada símbolo de función m-aria $f \in \mathcal{L}$, se tiene una función
$$f^{\mathfrak{A}} : A^m \to A$$
- Para cada símbolo de relación n-aria $R \in \mathcal{L}$, se tiene una relación
$$R^{\mathfrak{A}} \subseteq A^n$$

Los elementos $c^{\mathfrak{A}}$, $f^{\mathfrak{A}}$ y $R^{\mathfrak{A}}$ se llaman **interpretaciones** de sus símbolos respectivos.

Denotamos una $\mathcal{L}$-estructura como $\mathfrak{A} = \langle A, c^{\mathfrak{A}}, \ldots, f^{\mathfrak{A}}, \ldots, R^{\mathfrak{A}}, \ldots \rangle$

Semántica de LPO: variables

Definición

Para una $\mathcal{L}$-fórmula φ, se definen sus **variables libres** $VL(\varphi)$ según

- Si φ es atómica, entonces $VL(\varphi) = V(\varphi)$
- Si $\varphi = (\neg\psi)$, entonces $VL(\varphi) = VL(\psi)$
- Si $\varphi = (\psi_1 \star \psi_2)$, entonces $VL(\varphi) = VL(\psi_1) \cup VL(\psi_2)$
- Si $\varphi = (\exists x.\psi)$ o $\varphi = (\forall x.\psi)$, entonces $VL(\varphi) = VL(\psi) \setminus \{x\}$

Las variables libres **no están cuantificadas**

Semántica de LPO: variables

Definición

Para una $\mathcal{L}$-estructura $\mathfrak{A}$ con dominio A, se define una **asignación** σ como una función que para cada variable asigna un valor en A

Además, extendemos σ para dar valor a los $\mathcal{L}$-términos

- Si $t = c$ es un símbolo de constante, entonces $\hat{\sigma}(t) = c^{\mathfrak{A}}$
- Si $t = c$ es una variable, entonces $\hat{\sigma}(t) = \sigma(x)$
- Si $t = f(t_1, \ldots, t_n)$, entonces $\hat{\sigma}(t) = f^{\mathfrak{A}}(\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n))$

Por simplicidad usaremos σ en lugar de $\hat{\sigma}$

Semántica de LPO

Sean $\mathcal{L}$ un vocabulario, $\mathfrak{A}$ una $\mathcal{L}$-estructura con dominio A y σ una asignación para $\mathfrak{A}$

Definición

Decimos que $(\mathfrak{A}, \sigma)$ **satisface** una $\mathcal{L}$-fórmula, denotado como $(\mathfrak{A}, \sigma) \vDash \varphi$ si, y solo si,

- $\varphi := \ t_1 = t_2$ y $\sigma(t_1) = \sigma(t_2)$
- $\varphi := \ R(t_1, \ldots, t_n)$ y $(\sigma(t_1), \ldots, \sigma(t_n)) \in R^{\mathfrak{A}}$
- $\varphi := \ (\neg\psi)$ y $(\mathfrak{A}, \sigma) \nvDash \psi$
- $\varphi := \ (\psi_1 \vee \psi_2)$ y $(\mathfrak{A}, \sigma) \vDash \psi_1$ o $(\mathfrak{A}, \sigma) \vDash \psi_2$

Semántica de LPO

Definición

- $\varphi :=\ (\psi_1 \wedge \psi_2)$, $(\mathfrak{A}, \sigma) \vDash \psi_1$ y $(\mathfrak{A}, \sigma) \vDash \psi_2$
- $\varphi :=\ (\psi_1 \to \psi_2)$ y $(\mathfrak{A}, \sigma) \nvDash \psi_1$ o $(\mathfrak{A}, \sigma) \vDash \psi_2$
- $\varphi :=\ (\psi_1 \leftrightarrow \psi_2)$ y ambos $(\mathfrak{A}, \sigma) \vDash \psi_1$, $(\mathfrak{A}, \sigma) \vDash \psi_2$ o ambos $(\mathfrak{A}, \sigma) \nvDash \psi_1$, $(\mathfrak{A}, \sigma) \nvDash \psi_2$
- $\varphi = (\exists x.\psi)$ y existe $a \in A$ tal que $(\mathfrak{A}, \sigma[x/a]) \vDash \psi$, donde

$$\sigma[x/a](y) = \begin{cases} a & y = x \\ \sigma(y) & y \neq x \end{cases}$$

- $\varphi = (\forall x.\psi)$ y para todo $a \in A$ se tiene que $(\mathfrak{A}, \sigma[x/a]) \vDash \psi$

Si φ es oración, decimos simplemente $\mathfrak{A} \vDash \varphi$

¿Qué punto débil tiene LPO?

En lógica proposicional, el problema de satisfacibilidad es difícil, pero tenemos algoritmos para resolverlo

¿Cómo sabemos si una fórmula en LPO es satisfacible?

- Debemos encontrar alguna $\mathcal{L}$-estructura que la satisfaga
- ¿Cuál es nuestro espacio de búsqueda?
- ¿Podemos hacer esto para una fórmula arbitraria?

Hoy estudiaremos la complejidad de este problema

Playlist Unidad III y Orquesta

Playlist: LogiWawos #3

Además sigan en instagram:
@orquesta_tamen

Objetivos de la clase

- ☐ Conocer las versiones en LPO de satisfacibilidad y tautologías
- ☐ Demostrar la complejidad de los problemas asociados
- ☐ Comprender el concepto de conjunto definible en LPO
- ☐ Definir conjuntos e interpretar fórmulas como conjuntos definibles
- ☐ Comprender el concepto de isomorfismo de estructuras
- ☐ Usar el teorema de isomorfismo para comprender los límites del poder expresivo de LPO

Programa

Satisfacibilidad

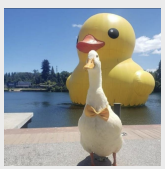
Definición

Una $\mathcal{L}$-fórmula φ es **satisfacible** si existe una $\mathcal{L}$-estructura $\mathfrak{A}$ y una asignación σ para $\mathfrak{A}$ tal que $(\mathfrak{A}, \sigma) \vDash \varphi$

Si φ es una $\mathcal{L}$-oración, entonces φ es satisfacible si existe $\mathfrak{A}$ tal que $\mathfrak{A} \vDash \varphi$

Ejercicio

Demuestre que la oración $\varphi = \forall x.P(x) \vee \forall x.\neg P(x)$ es satisfacible.



Fórmulas válidas

Definición

Una $\mathcal{L}$-fórmula φ es **válida** si para toda $\mathcal{L}$-estructura $\mathfrak{A}$ y toda asignación σ para $\mathfrak{A}$ se cumple que $(\mathfrak{A}, \sigma) \vDash \varphi$

Si φ es una $\mathcal{L}$-oración, entonces φ es válida si para toda $\mathfrak{A}$ se cumple que $\mathfrak{A} \vDash \varphi$

(Una $\mathcal{L}$-oración φ válida es **tautología** si se obtiene reemplazando fórmulas de LPO en variables proposicionales en una tautología de lógica proposicional)

Ejercicio

1. Muestre un ejemplo de oración válida
2. Muestre un ejemplo de oración válida que no sea tautología
3. Muestre que $\varphi = \forall x.P(x) \vee \forall x.\neg P(x)$ no es válida

Problemas de decisión en LPO

Al igual que en lógica proposicional, definimos problemas de decisión en LPO

$$\begin{aligned} \mathrm{SAT} &= \{\varphi \mid \varphi \text{ es una oración satisfacible}\} \\ \mathrm{VAL} &= \{\varphi \mid \varphi \text{ es una oración válida}\} \end{aligned}$$

¿Son igual de difíciles que en el caso proposicional?

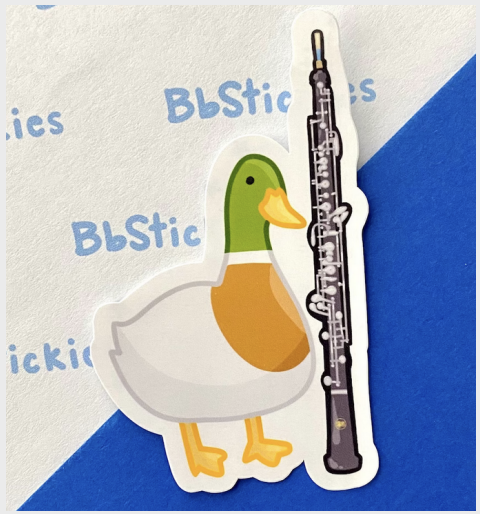
Programa

Complejidad de VAL

Teorema (Church)

VAL es indecidible

Demostración



Complejidad de VAL

Teorema (Church)

VAL es indecidible

Demostración

Consideramos el siguiente lenguaje **indecidible**

$$L_\epsilon \quad = \quad \{\mathcal{M} \mid \mathcal{M} \text{ acepta } \epsilon\}$$

y lo reduciremos a VAL.

¿Cómo se demuestra que L_ϵ es indecidible?

Complejidad de VAL

Demostración

Para cada máquina determinista $\mathcal{M}$, construiremos una oración $\varphi_{\mathcal{M}}$ en LPO tal que

$$\mathcal{M} \text{ acepta } \epsilon \quad \text{si, y solo si,} \quad \varphi_{\mathcal{M}} \text{ es válida}$$

Supondremos que $\mathcal{M} = (Q,\ \{0,1\},\ q_0,\ \delta,\ F)$ con

- $Q = \{q_0, \ldots, q_m\}$
- $F = \{q_m\}$
- no existe transición en δ para q_m

Nuevamente, estos supuestos son convenientes para facilitar $\varphi_{\mathcal{M}}$

Paréntesis: Complejidad de VAL

¿En qué se diferencia esta demostración a la del teorema de Cook?

- En Cook, la máquina era polinomial...
- ...teníamos una cota $t_{\mathcal{M}}(n)$ para el número de celdas y tiempo
- ¿En este caso sabemos algo del tiempo máximo que podría tardar $\mathcal{M}$ en aceptar ϵ?

Como L_ϵ es indecidible, no podemos acotar el tiempo de $\mathcal{M} \in L_\epsilon$

Paréntesis: Complejidad de VAL

Nuestro desafío:

- No sabemos el largo de la cinta *a priori*
- Codificaremos el tiempo con un número natural
- Codificaremos la posición como un entero

¿Cómo hacemos que la fórmula tome valores de esos dominios?

Complejidad de VAL

Demostración

Comenzamos definiendo nuestro vocabulario $\mathcal{L}$ y la motivación detrás de cada símbolo

$L(x, y)$	:	x es menor que y, representa el $<$
$S(x, y)$	:	y es el sucesor de x
$P(t)$	:	t es el tiempo de partida de la máquina
$C(t, p)$	:	$\mathcal{M}$ tiene un 0 en la posición p en tiempo t
$U(t, p)$	:	$\mathcal{M}$ tiene un 1 en la posición p en tiempo t
$B(t, p)$	:	$\mathcal{M}$ tiene un $\sqcup$ en la posición p en tiempo t
$E_i(t)$	:	$\mathcal{M}$ tiene estado q_i $(0 \leq i \leq m)$ en tiempo t
$H(t, p)$	:	la cabeza está en la posición p en tiempo t

Recordar que por sí mismos, los símbolos de $\mathcal{L}$ no tienen significado

Complejidad de VAL

Ahora modelamos las propiedades que queremos para los símbolos mediante fórmulas

Primero, queremos que L sea interpretado como un **orden total estricto**

$$\begin{aligned}\varphi_L \quad = \quad & \forall x.\ \neg L(x,x) \wedge \\ & \forall x.\forall y.\forall z.[(L(x,y) \wedge L(y,z)) \rightarrow L(x,z)] \wedge \\ & \forall x.\forall y.(x = y \vee L(x,y) \vee L(y,x)) \wedge \\ & \forall x.\exists y.(L(x,y) \wedge \neg\exists z.(L(x,z) \wedge L(z,y))) \wedge \\ & \forall x.\exists y.(L(y,x) \wedge \neg\exists z.(L(y,z) \wedge L(z,x)))\end{aligned}$$

φ_L obliga a tener estructuras donde L se interpreta como orden estricto y tales que existe sucesor y antecesor

Complejidad de VAL

Segundo, exigimos una relación de sucesor

$$\varphi_S \quad = \quad \forall x.\forall y.[S(x,y) \leftrightarrow (L(x,y) \wedge \neg\exists z.(L(x,z) \wedge L(z,y)))]$$

φ_S obliga a tener estructuras donde S se interpreta como la relación de sucesor clásica

Complejidad de VAL

Tercero, la máquina tiene un único punto de partida

$$\varphi_P \quad = \quad \exists x.[P(x) \wedge \forall y.(x \neq y \rightarrow \neg P(y))]$$

φ_P obliga a que un único tiempo corresponda al inicio de la ejecución

Cuarto, la máquina tiene un estado inicial

$$\varphi_I \quad = \quad \forall x.[P(x) \rightarrow (E_0(x) \wedge H(x,x) \wedge \forall y.B(x,y))]$$

φ_I obliga a que en el punto de partida
se cumplan las condiciones de inicialización

Complejidad de VAL

Quinto, la máquina funciona correctamente. Definimos cuatro subfórmulas que combinamos con conjunción en φ_C

Cada celda tiene un único símbolo siempre

$$\begin{aligned}\forall x.\forall y.\quad & [(C(x,y) \wedge \neg U(x,y) \wedge \neg B(x,y)) \vee \\ & (\neg C(x,y) \wedge U(x,y) \wedge \neg B(x,y)) \vee \\ & (\neg C(x,y) \wedge \neg U(x,y) \wedge B(x,y))]\end{aligned}$$

La máquina siempre está en un único estado

$$\forall x.\ \Big(\bigvee_{i=0}^{m}\Big(E_i(x) \wedge \bigwedge_{j=0,\ j\neq i}^{m} \neg E_j(x)\Big)\Big)$$

Complejidad de VAL

La cabeza siempre está en una única posición

$$\forall x.\exists y.[H(x,y) \wedge \forall z.(y \neq z \rightarrow \neg H(x,z))]$$

El contenido de una celda no cambia si no es apuntada por la cabeza

$$\begin{aligned}\forall x.\forall y.\forall z. \quad & (\neg H(x,y) \wedge S(x,z)) \rightarrow [(C(x,y) \wedge C(z,y)) \vee \\ & (U(x,y) \wedge U(z,y)) \ \vee \ (B(x,y) \wedge B(z,y))]\end{aligned}$$

Complejidad de VAL

Sexto, la función δ define cómo funciona la máquina

Para cada transición se define una fórmula y φ_δ es la conjunción de todas ellas

Por ejemplo, si $\delta(q_i, 0) = (q_j, 1, \triangleleft)$, se define la siguiente subfórmula

$$\forall x.\forall y.\forall u.\forall v. \quad [E_i(x) \wedge H(x,y) \wedge C(x,y) \wedge S(x,u) \wedge S(v,y)] \rightarrow (E_j(u) \wedge H(u,v) \wedge U(u,y))$$

Séptimo y final! La máquina acepta la palabra vacía

$$\varphi_A = \exists x.\exists y.\ P(x) \wedge (x = y \vee L(x,y)) \wedge E_m(y)$$

¿Cómo se define $\varphi_{\mathcal{M}}$ a partir de estas siete fórmulas?

Complejidad de VAL

Resumimos las oraciones

φ_L	:	L es orden estricto
φ_S	:	S es la relación de sucesor, respecto al orden L
φ_P	:	Hay un único instante de partida
φ_I	:	La configuración inicial es correcta
φ_C	:	La máquina funciona bien
φ_δ	:	La máquina funciona acorde a δ
φ_A	:	La máquina acepta la palabra vacía

Definimos

$$\varphi_{\mathcal{M}} := (\varphi_L \wedge \varphi_S \wedge \varphi_P \wedge \varphi_I \wedge \varphi_C \wedge \varphi_\delta) \to \varphi_A$$

¿Por qué se define con implicancia y no como una gran conjunción?

Complejidad de SAT

Corolario

SAT en LPO es indecidible

Demostración

Propuesta!

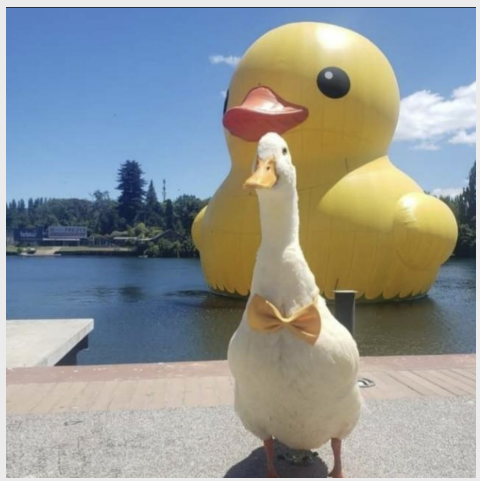
¡En LPO ni siquiera podemos dar un algoritmo para verificar satisfacibilidad!

Indecidibilidad en LPO

Ejercicio

Demuestre que el siguiente lenguaje es indecidible

$$\mathrm{EQUIV}_{=} \quad \{(\varphi, \psi) \quad | \, \varphi, \psi \ \mathcal{L}\text{-oraciones y para toda } \mathcal{L}\text{-estruc. } \mathfrak{A} \text{ se tiene que } \mathfrak{A} \vDash \varphi \text{ si, y solo si, } \mathfrak{A} \vDash \psi\}$$



Programa

Programa

El problema de definibilidad

Nos interesa trabajar un momento con fórmulas con variables libres (no oraciones)

¿Cuándo es verdadera una fórmula $\varphi(x_1, \ldots, x_k)$?

- Necesitamos una estructura $\mathfrak{A}$ y una asignación σ en $\mathfrak{A}$
- Cada variable libre debe ser **fijada**
- Lo denotamos por $(\mathfrak{A}, \sigma) \vDash \varphi(x_1, \ldots, x_k)$ en caso afirmativo

¿Qué sabemos de los valores que toman las variables libres en la asignación?

Nos acercaremos a comprender el poder expresivo de la LPO

El problema de definibilidad

Notación

Si $(\mathfrak{A}, \sigma) \vDash \varphi(x_1, \ldots, x_k)$ y la asignación es $\sigma(x_i) = a_i$ para cada $1 \leq i \leq k$, entonces denotamos

$$\mathfrak{A} \vDash \varphi(a_1, \ldots, a_k)$$

Problema de definibilidad

Dada una estructura $\mathfrak{A}$ con dominio A y $\mathcal{S} \subseteq A^k$ para $k \geq 1$, decimos que $\mathcal{S}$ es **definible** en $\mathfrak{A}$ si existe una fórmula $\varphi(x_1, \ldots, x_k)$ tal que

$$\mathcal{S} = \{(a_1, \ldots, a_k) \in A^k \mid \mathfrak{A} \vDash \varphi(a_1, \ldots, a_k)\}$$

¿Todo conjunto $\mathcal{S}$ es definible?

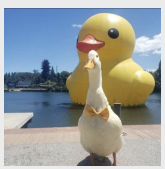
El problema de definibilidad

Ejercicio

Sea $\mathcal{L} = \{+, \cdot\}$, donde $+$ y $\cdot$ son símbolos de funciones binarias y considere $\mathfrak{A} = \langle \mathbb{N},\ +^{\mathfrak{A}},\ \cdot^{\mathfrak{A}} \rangle$ con la interpretación usual sobre los naturales

Identifique qué conjuntos definen en $\mathfrak{A}$ las siguientes $\mathcal{L}$-fórmulas

- $\varphi_1(x) = \forall y(x + y = y)$
- $\varphi_2(x) = \forall y(x \cdot y = y)$
- $\varphi_3(x, y) = \exists z(\neg\varphi_1(z) \wedge (x + z = y))$
- $\varphi_4(x, y) = \exists z(\varphi_2(z) \wedge (x + z = y))$



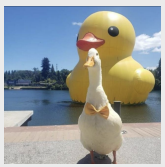
El problema de definibilidad

Ejercicio

Sea $\mathcal{L} = \{+, \cdot\}$, donde $+$ y $\cdot$ son símbolos de funciones binarias y considere $\mathfrak{A} = \langle \mathbb{N},\ +^{\mathfrak{A}},\ \cdot^{\mathfrak{A}} \rangle$ con la interpretación usual sobre los naturales

Demuestre que los siguientes conjuntos son definibles en $\mathfrak{A}$

- $\mathcal{S}_1 = \{a \in \mathbb{N} \mid a \text{ es un número primo}\}$
- $\mathcal{S}_2 = \{(a, b, c) \in \mathbb{N}^3 \mid a \equiv b \mod c\}$



El poder expresivo de LPO

Si queremos demostrar que un conjunto es **definible**

- Buscamos una fórmula adecuada que lo defina
- . . . puede ser difícil de construir

¿Todo conjunto $\mathcal{S}$ es definible en LPO?

El poder expresivo de LPO

Pati-Reflexión

Sea $\mathcal{L} = \{+\}$, donde $+$ es símbolo de función binaria. Sea $\mathfrak{A} = \langle \mathbb{R}, +^{\mathfrak{A}} \rangle$ con la interpretación usual de la suma sobre los reales

¿Se puede definir la multiplicación en $\mathfrak{A}$? Es decir, ¿es definible el siguiente conjunto $\mathcal{S}$?

$$\mathcal{S} = \{(a, b, c) \in \mathbb{R}^3 \mid a \cdot b = c\}$$

Para demostrar que un conjunto **no es definible en LPO**, necesitamos un resultado fundamental

Programa

Recordatorio: estructuras

Ejemplo

Consideremos $\mathcal{L} = \{0, 1, s, +, \cdot, <\}$. Los números naturales se pueden representar con la $\mathcal{L}$-estructura

$$\mathfrak{N} = \langle \mathbb{N},\ 0^{\mathfrak{N}},\ 1^{\mathfrak{N}},\ s^{\mathfrak{N}},\ +^{\mathfrak{N}},\ \cdot^{\mathfrak{N}},\ <^{\mathfrak{N}} \rangle$$

- $0^{\mathfrak{N}}$ es el "cero" de $\mathbb{N}$ y $1^{\mathfrak{N}}$ es el "uno" de $\mathbb{N}$
- $s^{\mathfrak{N}} : \mathbb{N} \to \mathbb{N}$ es una función unaria definida según
$$s^{\mathfrak{N}}(n) = n + 1$$
- $+^{\mathfrak{N}} : \mathbb{N}^2 \to \mathbb{N}$ es una función binaria definida según
$$+^{\mathfrak{N}}(a, b) = a + b$$
- $\cdot^{\mathfrak{N}} : \mathbb{N}^2 \to \mathbb{N}$ es una función binaria definida según
$$\cdot^{\mathfrak{N}}(a, b) = a \cdot b$$
- $<^{\mathfrak{N}} \subseteq \mathbb{N}^2$ es una relación binaria definida según
$$<^{\mathfrak{N}} = \{(a, b) \in \mathbb{N}^2 \mid a < b\}$$

Recordatorio: estructuras

Ejemplo

Consideremos otra $\mathcal{L}$-estructura para el mismo vocabulario

$$\mathfrak{A} = \langle A,\ 0^{\mathfrak{A}},\ 1^{\mathfrak{A}},\ s^{\mathfrak{A}},\ +^{\mathfrak{A}},\ \cdot^{\mathfrak{A}},\ <^{\mathfrak{A}} \rangle$$

- $A = \{\epsilon,\ 1,\ 11,\ 111,\ 1111\} = \{1\}^*$ (palabras de unos)
- $0^{\mathfrak{A}} = \epsilon$ y $1^{\mathfrak{A}} = 1$ (palabra "1")
- $s^{\mathfrak{A}} : A \to A$ es una función unaria definida según

$$s^{\mathfrak{A}}(w) = w1 \quad \text{(concatenar un 1)}$$

- $+^{\mathfrak{A}} : A^2 \to A$ es una función binaria definida según

$$+^{\mathfrak{A}}(w_1, w_2) = w_1 w_2 \quad \text{(concatenar las dos palabras)}$$

La noción de isomorfismo

La clase pasada presentamos dos estructuras que parecían relacionadas

- Ambas nos permitían representar propiedades en los naturales
- En cierto sentido, tienen **la misma forma**

¿Qué comparten? ¿Qué las diferencia?
¿Podemos abstraernos de esas diferencias?

La noción de isomorfismo

Sea un vocabulario $\mathcal{L}$ y dos estructuras $\mathfrak{A}$ y $\mathfrak{B}$ con dominios A y B respectivamente

Definición (isomorfismo)

Dos $\mathcal{L}$-estructuras $\mathfrak{A}$ y $\mathfrak{B}$ son **isomorfas**, denotado por $\mathfrak{A} \cong \mathfrak{B}$ si existe una biyección $h : A \to B$ tal que

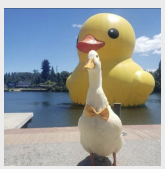
- para cada símbolo de constante $c \in \mathcal{L}$
$$h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$$
- para cada sím. de función m-aria $f \in \mathcal{L}$ y elementos $a_1, \ldots, a_m \in A$
$$h(f^{\mathfrak{A}}(a_1, \ldots, a_m)) = f^{\mathfrak{B}}(h(a_1), \ldots, h(a_m))$$
- para cada sim. relación n-aria $R \in \mathcal{L}$ y elementos $a_1, \ldots, a_m \in A$
$$(a_1, \ldots, a_n) \in R^{\mathfrak{A}} \text{ si, y solo si, } (h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$$

Llamamos a tal h un **isomorfismo** de $\mathfrak{A}$ en $\mathfrak{B}$

La noción de isomorfismo

Ejercicio

Sean $\mathfrak{N} = \langle \mathbb{N},\ 0^{\mathfrak{N}},\ 1^{\mathfrak{N}},\ +^{\mathfrak{N}},\ <^{\mathfrak{N}} \rangle$ y $\mathfrak{B} = \langle B,\ 0^{\mathfrak{B}},\ 1^{\mathfrak{B}},\ +^{\mathfrak{B}},\ <^{\mathfrak{B}} \rangle$, donde B es el conjunto de los naturales pares, $1^{\mathfrak{B}}$ se interpreta como el número 2 y las demás interpretaciones son las usuales. ¿Es cierto que $\mathfrak{A} \cong \mathfrak{B}$?



La noción de isomorfismo

Si dos estructuras $\mathfrak{A}$ y $\mathfrak{B}$ son isomorfas

- Entonces son **idénticas** excepto por sus dominios
- En cierto sentido, $\mathfrak{A}$ y $\mathfrak{B}$ son **indistinguibles**

En LPO es fundamental la imposibilidad de distinguir estructuras isomorfas

Teorema de isomorfismo (v1.0)

Teorema

Si $\mathfrak{A}$ y $\mathfrak{B}$ son $\mathcal{L}$-estructuras isomorfas, entonces para toda $\mathcal{L}$-oración φ se tiene que

$$\mathfrak{A} \vDash \varphi \quad \text{si, y solo si,} \quad \mathfrak{B} \vDash \varphi$$

Para poder demostrar que conjuntos no son definibles, necesitamos la versión más fuerte de este teorema

Teorema de isomorfismo (v2.0)

Teorema (isomorfismo)

Sean $\mathfrak{A}$ y $\mathfrak{B}$ $\mathcal{L}$-estructuras, σ asignación para $\mathfrak{A}$ y h un isomorfismo de $\mathfrak{A}$ en $\mathfrak{B}$. Entonces para toda $\mathcal{L}$-fórmula φ se tiene que

$$(\mathfrak{A}, \sigma) \vDash \varphi \quad \text{si, y solo si,} \quad (\mathfrak{B}, h \circ \sigma) \vDash \varphi$$

Observación

- $h \circ \sigma$ es una asignación para $\mathfrak{B}$

La versión v1.0 es un corolario de este teorema

Conjuntos no definibles

Ejemplo

Sea $\mathcal{L} = \{+\}$, donde $+$ es símbolo de función binaria. Sea $\mathfrak{A} = \langle \mathbb{R}, +^{\mathfrak{A}} \rangle$ con la interpretación usual de la suma sobre los reales

Demuestre que no se puede definir la multiplicación en $\mathfrak{A}$. Es decir, que el siguiente conjunto no es definible

$$\mathcal{S} = \{(a, b, c) \in \mathbb{R}^3 \mid a \cdot b = c\}$$

Usaremos el teorema de isomorfismo para obtener una contradicción

Conjuntos no definibles

Ejemplo

Demuestre que no se puede definir la multiplicación en $\mathfrak{A}$. Es decir, que el siguiente conjunto no es definible

$$\mathcal{S} = \{(a, b, c) \in \mathbb{R}^3 \mid a \cdot b = c\}$$

Supongamos que existe $\varphi(x, y, z)$ tal que para todo $a, b, c \in \mathbb{R}$

$$\mathfrak{A} \vDash \varphi(a, b, c) \quad \text{si, y solo si,} \quad a \cdot b = c$$

Luego, consideremos el isomorfismo de $\mathfrak{A}$ en $\mathfrak{A}$, $h : \mathbb{R} \to \mathbb{R}$ dado por $h(x) = x/2$.

Notamos que $\mathfrak{A} \vDash \varphi(2, 2, 4)$, pero $\mathfrak{A} \nvDash \varphi(h(2), h(2), h(4))$. Esto contradice el resultado del teorema de isomorfismo.

Concluímos que tal φ no existe, y por lo tanto $\mathcal{S}$ no es definible. □

Teorema de isomorfismo

Esta es la primera evidencia de que no podemos expresar **todo lo que queramos** en LPO

- Próxima clase demostraremos el teorema
- Exploraremos otras formas de definibilidad en LPO
- ... y estudiaremos sus alcances

Próximamente: teorema de compacidad

Programa

Objetivos de la clase

- ☐ Conocer las versiones en LPO de satisfacibilidad y tautologías
- ☐ Demostrar la complejidad de los problemas asociados
- ☐ Comprender el concepto de conjunto definible en LPO
- ☐ Definir conjuntos e interpretar fórmulas como conjuntos definibles
- ☐ Comprender el concepto de isomorfismo de estructuras
- ☐ Usar el teorema de isomorfismo para comprender los límites del poder expresivo de LPO

¿Qué aprendí hoy? ¿Comentarios?

Ve a

www.menti.com

Introduce el código

6449 2943

O usa el código QR